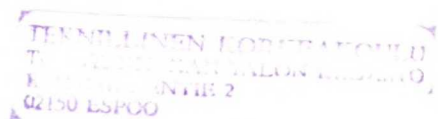


TEKNILLINEN KORKEAKOULU
Tietotekniikan osasto
Tietoliikenneohjelmistojen ja multimedian laboratorio

Jari Hautio

Multimedia ja palvelunlaatu langattomassa verkossa

Diplomityö
Marraskuu, 2002



Valvoja/Ohjaaja Professori Petri Vuorimaa

Tekijä: Jari Hautio

Työn nimi: Multimedia ja palvelunlaatu langattomassa verkossa

Päivämäärä: 6.11.2002

Sivuja yhteensä: 72

Osasto: Tietotekniikka

Professuuri: Vuorovaikutteinen digitaalinen media

Valvoja: Prof. Petri Vuorimaa

Multimedian ja langattomien verkkojen yleistyessä nousee tarve siirtää laajakaistaista multimedialiikennettä myös langattomassa verkossa. Langattoman verkon pieni kapasiteetti yhdessä epävarman radiotien kanssa aiheuttavat tarpeen kehittää tiedonsiirtoa langattomassa verkossa siten, että multimedian käyttö langattomassa verkossa on sujuvaa.

Palvelunlaatua ja multimediasovellusten tietoliikennetarpeita tutkimalla päädytään CBQ-jonotusta hyödyntävään ratkaisuun, jolla multimedian siirtoa langattomassa verkossa voidaan parantaa. Ratkaisun toimivuutta testataan simuloimalla useiden multimedia-sovellusten toimintaa käyttäen sekä perinteistä jonotusta että CBQ-pohjaista jonotusta.

Simulointien perusteella voidaan todeta CBQ-pohjaisen ratkaisun toimivan hyvin. Liikenteen priorisoinnilla ja kapasiteetin jaolla erityyppisten liikenneluokkien kesken oli kuitenkin ratkaiseva vaikutus toimivuuteen. Lisäksi heikon radioyhteyden päässä sijaitsevien käyttäjien havaittiin aiheuttavan ongelmia laadukkaan datan siirrolle.

Asiasanat: Palvelunlaatu, multimedia, langattomat verkot, CBQ, WLAN, simulointi

Author: Jari Hautio

Title: Multimedia and Quality of Service in Wireless Network

Date: 6.11.2002

Pages: 72

Department: Computer Science

Professorship : Interactive Digital Media

Supervisor: Prof. Petri Vuorimaa

New bandwidth consuming multimedia services press wireless networks to their limits. Providing high quality multimedia services in limited bandwidth wireless networks demands efficient multimedia aware transport methods.

The service needs of multimedia traffic are identified and quality of service enabled basestation is introduced. Strengths and weaknesses of the approach are further studied with simulations.

Results indicate that CBQ based system can improve performance of multimedia traffic in wireless networks. However, several weaknesses and limitations in the system are identified. Efficient communications require well tuned bandwidth allocation and poor radio channels can always cause unexpected behaviour in wireless networks.

Keywords: Quality of Service, multimedia, wireless networks, CBQ, WLAN, simulation

Alkulause

Tahdon kiittää professori Hannu Karia diplomityön aiheen innoittajana sekä professori Petri Vuorimaata ohjauksesta ja opastuksesta työn aikana.

Vaimoani Kaisaa haluan kiittää tuesta ja kannustuksesta sekä kielellisen ilmaisuuden huoltamisesta.

Espoossa 6.11.2002,



Jari Hautio

Lyhenneluettelo

BER	Bit Error Ratio
CBQ	Class Based Queueing
CSDPS	Channel state dependent packet switching
CSS	Cascaded Style Sheets
DHTML	Dynamic HTML
DRR	Deficit Round Robin
ECN	Explicit Congestion Notification
FFQ	Fluid Fair Queueing
FIFO	First in First out
FQ	Fair Queueing
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
HTML	Hypertext Markup Language
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
JPEG	Joint Picture Expert Group
MAC	Medium Access Control
MHEG	Multimedia Hypermedia Experts Group
MPEG	Motion Picture Expert Group
QCIF	Quarter Common Interchange Format
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RTP	Real-Time Transport Protocol
RTSP	RTP Signaling Protocol
SGML	Standard Generalized Markup Language
SMIL	Synchronized Multimedia Integration Language
TCP	Transport Control Protocol
UDP	User Datagram Protocol
WFQ	Weighted Fair Queueing
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	Extensible Markup Language

Sisällysluettelo

1 Johdanto.....	1
1.1 Työn rakenne.....	2
2 Tutkimusongelma.....	3
2.1 Aihepiirin esittely.....	3
2.2 Johdatus ongelmaan.....	5
2.3 Tutkimusongelma.....	7
3 Palvelunlaatu.....	9
3.1 Mitä on palvelunlaatu.....	9
3.2 Palvelunlaatu IP-verkoissa.....	9
3.3 Lankaverkoissa käytettävät jonotusalgoritmit.....	14
3.4 Palvelunlaadun kehysarkkitehtuurit.....	17
3.5 Palvelunlaatu langattomissa verkoissa.....	20
4 Multimedia Internetissä.....	23
4.1 Multimediasovelluksen rakenne.....	23
4.2 Multimedian siirtostandardit.....	25
5 Multimedian tehokas siirto langattomassa verkossa.....	37
5.1 Multimedian palvelutarpeet.....	37
5.2 Palvelutarpeiden tyydyttäminen.....	39
6 Simuloinnit langattomassa verkossa.....	40
6.1 Simulaatioiden tavoitteet.....	40
6.2 Testausympäristö.....	40
6.3 Multimedialiikenteen mallintaminen.....	47
6.4 Simuloinnit.....	52
7 Johtopäätökset.....	67
7.1 Työn jatkaminen.....	68

1 Johdanto

Uudet multimediasovellukset asettavat aikaisempaa tiukempia vaatimuksia tietoverkoille. Tyypillisesti nämä sovellukset on suunniteltu nopeisiin lankaverkkoihin, joissa on tarpeeksi kapasiteettia kattamaan sovellusten tarpeet. Langattomien verkkojen yleistyessä lisääntyy tarve käyttää multimediasovelluksia myös langattomissa verkoissa, joissa kaistanleveys on huomattavasti pienempi kuin lankaverkoissa. Sovellukset voivat sopeutua pienempään kaistanleveyteen käyttämällä tehokkaampia tiedonpakkausalgoritmeja, mutta pakkaustiheyden kasvattaminen heikentää multimediaesityksen kuvan- ja äänenlaatua. Toinen menetelmä on kehittää langattomassa verkossa tapahtuvaa multimediatatan siirtoa siten, että verkko pystyy siirtämään multimediatataa mahdollisimman tehokkaasti.

Multimedialiikenne muodostaa suhteellisen tasaisen datavirran, joka vaatii esteettömän kulun verkon halki palvelimelta asiakkaalle. Kapeikot ja katkokset siirtotieässä viivästävät virtaa ja voivat näkyä virheinä multimediaesityksessä. Jos datavirran ja siirtotien leveys on sovellusten tiedossa, ne voivat sopeuttaa liikenteen määrän vastaamaan saatavilla olevaa kaistanleveyttä. Verkon liikennemäärien jatkuva vaihtelu kuitenkin vaikeuttaa sovellusten tehtävää, ja ilman verkon apua ne joutuvat reagoimaan havaittuihin ongelmiin, mutta eivät pysty estämään ongelmien syntyä. Tehokas multimediatatan siirto edellyttää, että verkko neuvottelee verkon käytöstä sovellusten kanssa ja tunnistaa multimedialiikenteen tarpeet sekä ohjaa verkon liikennettä siten, että multimedialiikenteen tarpeet tulevat tyydytetyiksi.

Tilanteissa, joissa verkko ei enää kykene tyydyttämään multimedialiikenteen vaatimuksia, sen on kyettävä hallitusti rajoittamaan liikenteen määrää siten, että ainakin osa multimediayhteyksistä säilyy käyttökelpoisina. Tämä voi tarkoittaa esimerkiksi sitä, että jokaisen yhteyden leveyttä rajoitetaan ja sovellukset joutuvat sopeutumaan uusiin kapeampiin siirtoväyliin.

Multimedialiikenteen erityiskohtelu ei kuitenkaan saa estää perinteistä dataa siirtävien sovellusten toimintaa kohtuuttomasti. Laajakaistainen multimedialiikenne voi helposti kuluttaa verkon kapasiteetin kokonaan ja estää perinteisten sovellusten toiminnan. Vaikka perinteisellä dataliikenteellä ei ole erityisiä siirtovaatimuksia, dataa siirtävät sovellukset odottavat, että edes jonkin verran dataa pääsee kohtuullisessa ajassa perille.

Tässä työssä haetaan ratkaisua siihen, miten multimediadataa voidaan tehokkaasti siirtää langattomassa verkossa aiheuttamatta kohtuuttomia sivuvaikutuksia perinteisen datan siirtoon. Tiedonsiirron kehittämisessä keskitytään verkkotason ratkaisujen löytämiseen ja sovellustason menetelmiä sivutaan vain kartoitettaessa multimedian vaatimuksia siirtotielle. Työssä rajoitutaan tarkastelemaan yhden tukiaseman ja useiden käyttäjien muodostamaa langatonta verkkoa.

1.1 Työn rakenne

Kappaleessa 2 perehdytään tutkimusongelman aihepiiriin ja esitellään kriteerit, joilla ongelman ratkaisua voidaan arvioida. Seuraavaksi tarkastellaan palvelunlaatua ja sen soveltamista langattomiin verkkoihin kappaleessa 3. Kappaleessa 4 tutustutaan keskeisimpiin multimedian sovelluskohteisiin ja multimediadatan siirtämiseen. Kappaleessa 5 esitellään palvelunlaatuun perustuva ratkaisu, jonka toimivuutta testataan simuloinneilla kappaleessa 6. Lopuksi esitellään työn tulokset ja mietitään työn jatkamista kappaleessa 7.

2 Tutkimusongelma

2.1 Aihepiirin esittely

Multimedia tarkoittaa vuorovaikutteista viestintää, jossa useat media yhdistyvät. Tyypillisesti tämä tarkoittaa kuvan, äänen ja videon synkronoitua käyttöä sovelluksessa. Tässä työssä *multimediasovelluksilla* viitataan sellaisiin sovelluksiin, jotka käyttävät suurta tiedonsiirtokapasiteettia vaativia medioita, kuten ääntä ja videota. Kokonaisen ääni- tai videotiedoston siirto Internetissä ennen esityksen alkua on hyvin hidasta. Sen sijaan esitettävää dataa siirretään esityksen aikana siten, että uutta dataa saapuu esityksen edetessä samaan tahtiin kuin sitä käytetään. Tästä tekniikasta käytetään nimitystä *virtaavaa dataa (streaming data)*. Tässä työssä termillä *multimediatdata* tarkoitetaan, ellei toisin erityisesti mainita, nimenomaan sellaista multimediatdataa, jota siirretään käyttämällä *virtaavaa dataa*. [1]

Virtaavaa dataa käyttävä asiakassovellus voi aloittaa datan esittämisen ennen kuin koko tiedosto on siirtynyt asiakkaan koneelle. Toisin sanoen käyttäjä voi aloittaa videon katselemisen vain pienellä viiveellä. *Virtaava data* tarvitsee vakaan tietoliikenne-yhteyden toimiakseen kunnolla, koska virtaavan datan uudelleen lähettäminen ei ole mahdollista. Yksittäisten pakettien katoaminen voidaan yleensä kompensoida sovellustasolla käyttämällä erilaisia virheenpiilotustekniikoita, mutta useiden peräkkäisten pakettien katoaminen aiheuttaa häiritsevän virheen esitykseen. [1]

Internettiä ei alun perin suunniteltu datavirtojen kuljetukseen, vaan tavoitteena oli yksittäisten pakettien siirtäminen. Internetin arkkitehtuuri perustuu yksittäisten lähde- ja kohdeosoitteen sisältävien *datasähkeitten* itsenäiseen liikkumiseen *reitittimien* muodostamassa verkossa. Mikäli johonkin verkon osaan kohdistuu enemmän liikennettä kuin verkko pystyy välittämään, reitittimet eivät pysty välittämään sinne enää uusia paketteja, vaan joutuvat hylkäämään saapuneen datasähkeen.

Internet Protokolla (IP) määrittelee osoitteenmuodostuksen, jonka avulla datasähkeitä voidaan käsitellä verkossa erillisinä paketteina. Tarvittaessa IP osaa myös jakaa datasähkeitä reitittimissä ja koota ne uudelleen vastaanottajassa. *IP* ei kuitenkaan tarjoa luotettavaa tiedonsiirtoa, vaan reitittimet voivat pudottaa datasähkeitä ilmoittamatta siitä lähettäjälle tai vastaanottajalle. Ylemmän kerroksen siirtoprotokollan, kuten TCP:n, tehtävänä on seurata datasähkeitten siirtoa ja tarvittaessa lähettää datasähke uudelleen. Mikään em. tiedonsiirtoprotokollista ei kuitenkaan pysty takaamaan datasähkeitten

saapumista ajallaan tai antamaan takuita datan läpimenosta. IP siis tarjoaa ns. paras yritys-palvelun, joka ei takaa datan saapumista perille tiettyä aikana eikä läpimenevän datan määrää. Multimediasovellusten kannalta tämä tarkoittaa, että IP-pohjainen Internet ei pysty tarjoamaan katkeamatonta virtaavan datan siirtoa, vaan sovellukset joutuvat varautumaan siihen, että tiedonsiirrossa voi tulla katkoksia. [2]

Multimediatatan siirtoprotokollan tehtävänä on toimittaa mahdollisimman tehokkaasti dataa lähettäjältä vastaanottajalle. *User Datagram Protocol (UDP)* on kevyt tiedonsiirtoprotokolla, joka on tarkoitettu datasähkeiden lähettämiseen sovellusten välillä. Se tarjoaa ainoastaan tarkistussumman laskennan ja multipleksauksen. Keveytensä puolesta se soveltuu multimediatatan tarpeisiin. [3]

Virtaavan datan siirtoon Internetissä käytetty perusprotokolla on *Real-Time Transport Protocol (RTP)*, joka määrittelee tavan lähettää reaaliaikaista dataa Internetissä. Protokollan tarjoamiin palveluihin kuuluvat hyötykuorman tyypin tunnistus, numerointi, aikaleimaus ja siirron seuranta. Normaalisti RTP-protokollaa ajetaan UDP:n päällä, jolloin voidaan hyödyntää UDP:n tarjoamia palveluita. [4]

Jonotusalgorithmi on reitittimissä käytetty algoritmi, joka valitsee, missä järjestyksessä paketit lähetetään edelleen ja mitkä paketit pudotetaan. Tyypillisesti Internetissä käytetään yksinkertaista jonotusalgorithmia (*DropTail*), jossa tulevat paketit ohjataan jonon perään ja jonoa puretaan alusta käsin. Jos jonolle varattu puskuritila loppuu, saapuvat paketit pudotetaan. Jonotusalgorithmi käsittelee jokaista pakettia erillisenä eikä takaa resurssien tasapuolista jakoa eri yhteyksien välille.

Palvelunlaadulla viitataan tietoliikenteessä verkon kykyyn antaa sovelluksille jonkinasteisia takeita tietoliikenneyhteyden laadusta. Tärkeimmät laatuparametrit ovat viive, viiveenvaihtelu, kaistanleveys ja pakettikato. Viiveellä tarkoitetaan aikaa, joka kuluu tiedonsiirtoon lähettäjältä vastaanottajalle. Viiveenvaihtelulla kuvataan vaihtelua, joka tapahtuu pakettien saapumisajoissa verrattuna niiden lähetysaikoihin. Kaistanleveydellä mitataan tiedonsiirtokanavan teoreettista maksimitiedonsiirtokapasiteettia ja pakettikato ilmoittaa, miten suuri osa lähetetyistä paketeista katoaa matkalla vastaanottajalle.

Langattomat lähiverkot käyttävät langatonta siirtomediaa, kuten radioaaltoja tai infrapunavaloa. Käyttäjät liittyvät verkkoon langattomien *yhteyspisteiden* kautta. Näistä

yhteyspisteistä käytetään radioverkoissa myös nimitystä *tukiasema*. Yksi yleinen langattomissa verkoissa käytetty standardi on IEEE 802.11 Ethernet -standardiin [5] perustuva WLAN. WLAN -verkon tarjoamat yhteysnopeudet 2.4 Ghz taajuusalueella ovat 1, 2, 5.5 ja 11Mb/s. Korkea taajuus rajoittaa yhteyspisteiden kantoaluetta ja yleensä langattomissa verkoissa yhteyspisteen kantoalue jää alle 500 metriin. Sisätiloissa kantoalue on huomattavasti pienempi, koska mm. seinät vaimentavat radioaaltoja. Yhteyspisteen kantoalueen rajaamaa aluetta kutsutaan *soluksi*. Langaton verkko muodostuu joukosta soluja, jotka sijaitsevat katvealueiden välttämiseksi osittain päällekkäin. Pieni solukoko pakottaa liikkuvat käyttäjät vaihtamaan usein yhteyspistettä, eli tekemään ns. handoffin. IEEE 802.11 -standardi määrittelee linkkitason protokollan handoffin suorittamiseksi samassa aliverkossa olevien yhteyspisteiden välillä, jolloin siirtyminen tapahtuu piilossa käyttäjältä. Eri verkkojen välillä tapahtuva sujuva liikkuminen voidaan hoitaa esimerkiksi *Mobile IP*:llä. Esteistä ja häiriöistä johtuen radioverkko on huomattavasti herkempi virheille kuin lankaverkko. IEEE 802.11 -standardin käyttämä linkkitason protokolla pyrkii kompensoimaan radiotien aiheuttamia virheitä.

2.2 Johdatus ongelmaan

Multimedian siirto tarvitsee vakaata tietoliikenneyhteyttä. Koska IP-pohjaisissa verkoissa yksittäisen yhteyden saamaa palvelua ei pystytä takaamaan, on multimedian siirtoon tyypillisesti varauduttu yksinkertaisesti lisäämällä tiedonsiirtokapasiteettia. Ylimoitettu tiedonsiirtokapasiteetti riittää palvelemaan multimedian tarpeita suurimman osan ajasta. Kapasiteetin lisääminen tarjoaa kuitenkin vain tilapäisen ratkaisun. Paremmat yhteydet lisäävät multimedian käyttöä ja ennen pitkää suurikaan kapasiteetti ei enää riitä lisääntyneen multimedian siirtoon.

Langattomissa verkoissa radiotien fyysiset ominaisuudet rajoittavat käytettävissä olevaa kaistanleveyttä. Yleensä kaistanleveys jää muutamiin megabitteihin sekunnissa, joka riittää yhden MPEG-2 -tasoisen videon katseluun. Toisen käyttäjän ilmaantuessa paikalle tiedonsiirtokapasiteetti loppuu helposti ja videon katselu häiriintyy. Tarvitaan siis ratkaisu, joka pureutuu ongelman ytimeen: multimedian vaatimusten ja IP-pohjaisen verkon tarjoamien palveluiden vastaamattomuuteen.

Langattomassa verkossa liikkuva data voidaan jakaa useisiin luokkiin sen mukaan miten tärkeää datan oikea-aikainen toimitus on. Perinteiset Internet-sovellukset, kuten FTP,

WWW ja e-mail, on suunniteltu toimimaan epäluotettavassa Internetissä. Ne eivät aseta vaatimuksia pakettien toimitusajalle tai perille pääsulle. Koska näissä sovelluksilla toimitusajankohta ei ole kriittinen sovellusten toiminnan kannalta, niiden siirtämä data luokitellaan *ei-aikakriittiseksi dataksi*.

Multimediasovelluksille pakettien satunnainen toimitus ei riitä, koska virtaavaa dataa ei ehditä lähettämään uudelleen ennen kuin sitä tarvitaan. Jokaisella paketilla on tietty esitysajankohta, johon mennessä niiden oltava perillä. Koska paketteja esitetään sitä mukaa kuin ne saapuvat, liikaa myöhästynyttä pakettia ei voida odottaa vaan se joudutaan hylkäämään. Muutamien yksittäisten pakettien putoaminen on hyväksyttävää, mutta usean peräkkäisen paketin putoaminen, kuten ylikuormitusilanteessa tyypillisesti käy, aiheuttaa väistämättä havaittavan virheen multimediaesitykseen. Puskuroinnilla voidaan kompensoida pieniä viiveen vaihteluita, mutta silloinkin viiveen vaihtelu on pysyttävä ennakoimaan. Pakettien toimitusajankohdalla on kriittinen vaikutus sovellusten toimintaan, joten multimedidata luokitellaan *aikakriittiseksi dataksi*.

Multimedidataa ja perinteistä dataa tehokkaasti siirtävän verkon perusedellytys on, että se huomioi erilaisten sovellusten erilaiset palvelutarpeet. Verkon on annettava etusija aikakriittiselle datalle perinteisen ei-aikakriittisen datan kustannuksella, mutta samalla verkon on huolehdittava siitä, että myös perinteisten sovellusten toimintakyky säilyy ja että ne saavat osansa kaistanleveydestä kohtuullisella viiveellä.

Radioverkoissa käyttäjän sijainti suhteessa yhteyspisteeseen vaikuttaa oleellisesti käyttäjälle tarjolla olevaan palvelutasoon. Tarkastellaan tilannetta ensin yhden käyttäjän näkökulmasta.

Kun käyttäjä pysyttelee reilusti kantoalueen sisäpuolella, käyttäjälle tarjolla oleva maksimikaistanleveys, viive ja viiveen vaihtelu säilyvät suhteellisen vakiona. Samalla pakettikato langattomassa verkossa pysyy käyttäjän kannalta nollassa. Kun käyttäjä siirtyy kantoalueen reunaa kohti, radiosignaali heikentyy ja virheiden todennäköisyys lisääntyy. WLAN-verkoissa käytetään linkkitason uudelleenlähetyksiä piilottamaan radiotien aiheuttamat virheet ylemmiltä kerroksilta. Tällöin huono radioyhteys näkyy käyttäjälle viiveen ja viiveen vaihtelun kasvuna sekä kaistanleveyden pienenemisenä. Kun kaistanleveys laskee liian pieneksi radioverkko ei enää ehdi korjaamaan pakettikatoja ja käyttäjä huomaa pakettikadon kasvun.

IEEE 802.11 -standardin mukaisessa verkossa päätelaitteet pystyvät sopeutumaan heikentyneeseen radioyhteyteen laskemalla yhteysnopeutta portaittain vastaanotetun signaalin heikentyessä. Alemmilla yhteysnopeuksilla voidaan käyttää koodausmenetelmiä, jotka sietävät paremmin virheitä. Tällöin virhetodennäköisyys laskee ja läpimenevän datan määrä nousee.

Kaikki saman solun alueella olevat käyttäjät jakavat yhteyspisteen tarjoaman kaistanleveyden ja kilpailevat samoista resursseista. Käyttäjien liikkuvuuden ja radiotien epävarmuuden takia absoluuttisten takeiden antaminen kaistanleveydestä, viiveestä, viiveen vaihtelusta ja pakettikadosta WLAN-verkossa on käytännössä mahdotonta.

2.3 Tutkimusongelma

Tämän työn tarkoituksena on tutkia, miten multimedialiikenteen siirtoa langattomassa verkossa voidaan tehostaa palvelunlaadun avulla. Työssä tunnistetaan multimedialiikenteen keskeiset palvelutarpeet ja tarkastellaan, miten erilaisilla palvelunlaatuun perustuvilla ratkaisuilla voidaan vaikuttaa multimedian siirtoon.

2.3.1 Ongelman rajaus

Työssä keskitytään tarkastelemaan erityisesti langattomassa yhteyspisteessä käytettävien jonotusalgoritmien vaikutusta liikenteeseen. Suuri osa multimedialiikenteestä suuntautuu palvelun käyttäjää kohti, joten yhteyspisteen toimintaa tehostamalla voidaan nostaa kokonaispalvelunlaatua langattomassa verkossa.

Useiden tukiasemien muodostama langaton verkko on palvelunlaadun kannalta erittäin monimutkainen kokonaisuus, joka vaatii sekä toimivan tiedonsiirron yhden yhteyspisteen sisällä, että sujuvat siirrot muiden yhteyspisteiden alueelle. Tässä työssä rajoitutaan tutkimaan perustapausta, jossa liikennöinti tapahtuu kokonaan yhden solun sisällä.

2.3.2 Ratkaisun kriteerit

Ratkaisun kriteereiden määrittelemisen aloitetaan tarkastelemalla ideaalista ratkaisua. Ideaalinen ratkaisu kuvaa järjestelmää, johon reaali maailman fyysiset rajoitteet eivät vaikuta. Ideaalisesta ratkaisusta voidaan tunnistaa kohdat, joissa fyysisen maailman rajoitteet estävät ideaalisen ratkaisun toteutumisen. Tämän pohjalta voidaan edelleen valita ne kriteerit, joilla ongelman ratkaisua on mielekkäintä arvioida.

Ideaalisessa tapauksessa radioverkon läpi kulkeva multimedia pystyy kulkemaan radiotien ylitse virheettömänä ja viiveettömänä. Tämä edellyttää kolmea asiaa. Ensinnäkin radiotiellä on oltava tarpeeksi kapasiteettia kuljettamaan kaiken siellä liikkuvan liikenteen. Toiseksi liikenteen on siirryttävä radiotiellä virheettömästi huolimatta radiohäiriöistä tai etäisyydestä tukiasemaan. Kolmanneksi saman tukiaseman alueella olevien käyttäjien liikenne ei saa vaikuttaa toisiinsa.

Radiotien kaistan rajoittamisesta seuraa väistämättä tilanteita, joissa radiotien kapasiteetti ei riitä kuljettamaan kaikkea liikennettä. Hetkellisistä ylikuormitustilanteista voidaan selvitä puskuroimalla liikennettä. Puskurointi lisää viivettä ylikuormitustilanteen aikana ja aiheuttaa suuremman viiveenvaihtelun liikenteeseen. Pitempään jatkuvan ylikuormitustilanteen aikana osa paketeista on pakko hylätä, jolloin virheiden määrä kasvaa.

Kuten edellä WLAN-verkon tarkastelun yhteydessä todettiin, radiohäiriöitä ja etäisyyden kasvua voidaan torjua käyttämällä vikasietoisempaa koodausta, mikä lisää virheettömyyden osuutta verkon kapasiteetista. Voimakkaat häiriöt ja suuri etäisyys kuitenkin aiheuttavat välttämättä myös pakettikatoja ja virheitä liikenteeseen, joten radiotien aiheuttamaa satunaisuutta on mahdoton piilottaa kokonaan käyttäjiltä.

Kaikki saman tukiaseman alueella liikennöivät käyttäjät jakavat saman siirtotien ja joutuvat siten jakamaan käytössä olevan kaistanleveyden keskenään. Kun yksi käyttäjä liikennöi tukiaseman kanssa, muut joutuvat odottamaan vuoroaan. Tästä aiheutuu välttämättä ainakin pieniä viiveitä ja viiveenvaihteluita. Mikäli käyttäjien kaistankäyttöä ei mitenkään valvota, voi seurata tilanne, jossa yksi käyttäjä hallitsee koko verkkoa, eivätkä muut käyttäjät pysty liikennöimään lainkaan.

Edeltävistä tarkasteluista voidaan edelleen tiivistää kaksi keskeistä kriteeriä, jonka mukaan tutkimusongelman ratkaisua voidaan arvioida: multimedialiikenteen saama palvelunlaatu ja käyttäjien saaman palvelun tasapuolisuus ylikuormitustilanteessa.

3 Palvelunlaatu

3.1 Mitä on palvelunlaatu

Yleisesti palvelunlaatu tarkoittaa palvelun käyttäjän tyytyväisyyttä saamaansa palveluun. Palvelunlaatu on käyttäjän subjektiivinen mielipide, johon vaikuttavat palvelukyky, palvelun helppokäyttöisyys, palvelun varmuus sekä muut palvelulle tyypilliset ominaisuudet.

Tietoliikenteessä palvelunlaadulla tarkoitetaan verkon kykyä tarjota sovelluksille jonkinlainen takuu siitä, että verkko pystyy täyttämään sovellusten tietoliikenne- ja palvelutarpeet. Palvelunlaadun muodostumisessa jokainen lähettäjän ja vastaanottajan välissä oleva verkkoelementti vaikuttaa omalta osaltaan tarjotun palvelun tasoon. Reitin heikoin lenkki määrää, minkä tasoista palvelunlaatua verkko pystyy tarjoamaan. [6]

Koska palvelunlaatu on suhteellisen laaja ja epämääräinen käsite, palvelunlaadun käsitettä on syytä rajata helpommin käsiteltävälle tasolle. Palveluluokkien (*differentiated Classes of Service*) lähtökohta on liikenteen ja palveluiden jakaminen luokkiin. Luokkien erilaisella kohtelulla voidaan tarjota osalle asiakkaista parempaa palvelua kuin toisille. Jaottelun tekemiseksi tarvitaan menetelmä, jolla eri luokkiin kuuluva liikenne voidaan tunnistaa. Jaottelu voidaan suorittaa joko liikenteen saapuessa verkkoon tai verkon sisällä. Tyypillisesti liikenteen jaotteluun käytetään mm. protokollaa, lähde- tai kohdeosoitteita tai virtatunnistetta. Näiden avulla voidaan liikenne jakaa esimerkiksi WWW-liikenteeseen, FTP-liikenteeseen ja multimedialiikenteeseen. [7]

3.2 Palvelunlaatu IP-verkoissa

Internetin IP-pohjainen arkkitehtuuri, jossa paketit toimitetaan ainoastaan osoitteiden perusteella, tarjoaa ns. paras yritys -palvelun. Tämä palvelutaso soveltuu ainoastaan ei-aikakriittistä dataa siirtävien sovellusten tarpeisiin, koska ylikuormitustilanteessa liikenteen siirtyminen on yksittäisen sovelluksen kannalta satunnaista. Aikakriittistä dataa siirtävien sovellusten palvelutarpeiden täyttämiseksi verkon on kyettävä tarjoamaan parempia takeita datan perillepääsystä.

Palvelunlaadun nostamisen perustana IP-verkoissa on liikenteen eriarvoinen kohtelu. Palveluluokkien avulla voidaan osalle liikenteestä tarjota parempaa palvelua muun liikenteen kustannuksella. Palveluluokkien määrittämiseksi tarvitaan mittareita, jotka kuvaavat palveluluokkien ominaisuuksia. Tärkeimmät palvelunlaadun määrittämiseen käytetyt mittarit ovat seuraavat:

- **Läpimeno** ilmoittaa, miten paljon asiakassovelluksen dataa siirtyy tietoliikente-yhteyden läpi. Läpimenon mittarina käytetään yleensä kilobittejä sekunnissa (kbps) tai Megabittejä sekunnissa (Mbps).
- **Viive** ilmoittaa siirtoviiveen, joka kuluu paketin siirtämiseen lähettäjältä vastaanottajalle.
- **Huojuunta** ilmoittaa viiveen vaihtelun, joka ilmenee esimerkiksi pakettivirran pakettien välisissä saapumisajoissa.
- **Pakettikato** ilmoittaa, miten suuri osa lähetetyistä paketeista katoaa matkalla lähettäjältä vastaanottajalle.
- **Saatavuus** ilmoittaa, miten suuren osan ajasta asiakas pystyy saamaan tilaamaansa palvelunlaatua.

Tarvittaessa palvelunlaadun määrittelyä voidaan tarkentaa lisämittareilla, jotka ottavat siirtyvän datan luonteen huomioon. Tällaisia mittareita ovat mm. suurin mahdollinen lähetyspurskeen koko ja suurin mahdollinen lähetysnopeus.

3.2.1 Palvelunlaadun toteuttaminen

Palvelunlaadun toteuttamiseksi IP-pohjaisissa verkoissa on tarjolla useita arkkitehtuurisia perusratkaisuja:

Ylikapasiteetti

IP-pohjaisen verkon palvelunlaadun puute ilmenee erityisesti verkon käyttöasteen noustessa lähelle verkon maksimikapasiteettia. Ilmeinen ratkaisu palvelunlaatuongelmien parantamiseksi on verkon kapasiteetin lisääminen. Verkon ruuhkautuminen voidaan estää siten, että verkko mitoitetaan maksimikuormituksen mukaan. Tämä ei kuitenkaan usein ole houkutteleva ratkaisu, koska tarvittavan maksimikapasiteetin arviointi etukäteen on

yleensä mahdotonta. Lisäksi se ei aina ole teknisesti mahdollista eikä isommassa mittakaavassa taloudellisesti järkevä vaihtoehto. Pienissä lankaverkoissa kapasiteetin lisäämisellä voidaan kuitenkin saavuttaa hyviä tuloksia verrattuna monimutkaisempien järjestelmien käyttöönottoon sisältyviin vaikeuksiin.

Resurssien varaaminen

Resurssinvarausjärjestelmän avulla verkko pystyy hallitsemaan siihen saapuvaa liikennettä ja huolehtimaan siitä, ettei liikenteen määrä ylitä verkon kapasiteettia. Resurssinvaraus perustuu varausviestien käyttöön. Viestien avulla sovellukset pyytävät verkolta kapasiteettia liikenteelleen. Mikäli verkossa on riittävästi kapasiteettia sovelluksen käyttöön, verkko varaa sovellukselle osan kapasiteetista ja antaa sille liikennöintiluvan. Kun koko kapasiteetti on varattu, uusia varauksia ei enää voida hyväksyä ja sovellusten on odotettava resurssien vapautumista. Menetelmän tehokkuus perustuu siihen, että kun verkkoon pääsevän liikenteen määrä rajoitetaan tarjolla olevan kapasiteetin mukaan, voidaan ylikuormitustilanteilta välttyä ja palvelunlaatua nostaa. Järjestelmässä käytetty pääsypolitiikka puolestaan tarjoaa mahdollisuuden määritellä, keillä on oikeus käyttää verkkoa. Normaalikuormituksella kaikille käyttäjille sallitaan pääsy verkkoon, mutta kun kuormitustaso nousee tarpeeksi korkealla, voidaan pääsyä rajoittaa ja resurssinvaraus sallia ainoastaan etuoikeutetuille käyttäjille. Integrated Services [8] on IETF:n kehittämä resurssinvaraukseen pohjautuva arkkitehtuuri.

Priorisointi

Priorisointijärjestelmässä sovellusten liikenne jaetaan eri luokkiin sen mukaan, minkä tasoista palvelua ne vaativat. Eri luokat asetetaan tärkeysjärjestykseen, ja tärkeämpien luokkien liikenne saa etusijan alemman luokan liikenteeseen nähden. Puhtaaseen priorisointiin perustuvan järjestelmän heikkoutena on alemman prioriteetin sovellusten nälkiintyminen tilanteessa, jossa ylemmän luokan liikenne riittää käyttämään koko verkon kapasiteetin. Kun luokille asetetaan minimikaistanleveydet, voidaan perinteisten sovellusten toimivuus taata myös pitkään jatkuvissa ylikuormitustilanteissa. Differentiated Services on yksi IETF:n kehittämä priorisointiin pohjautuva arkkitehtuuri.

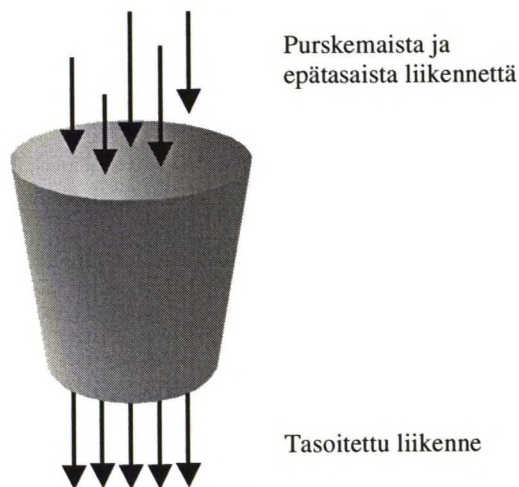
3.2.2 Liikenteen ohjausmenetelmät

Palvelunlaadun toteutus edellyttää, että verkolla on käytössään tehokkaita menetelmiä, joilla verkossa kulkevaa liikennettä voidaan ohjata. Yleisimmät liikenteenohjaukseen käytetyt välineet ovat liikenteen muokkaus, pääsyvalvonta, jonotusalgoritmit sekä ruuhkautumisen hallinta. [7]

Liikenteen muokkaus

Liikenteen muokkaus on mekanismi, jolla verkkoon lähetettävän liikenteen määrää ja lähetysnopeutta voidaan hallita. Yleisimmät muokkausmekanismit ovat *leaky bucket* ja *token bucket*.

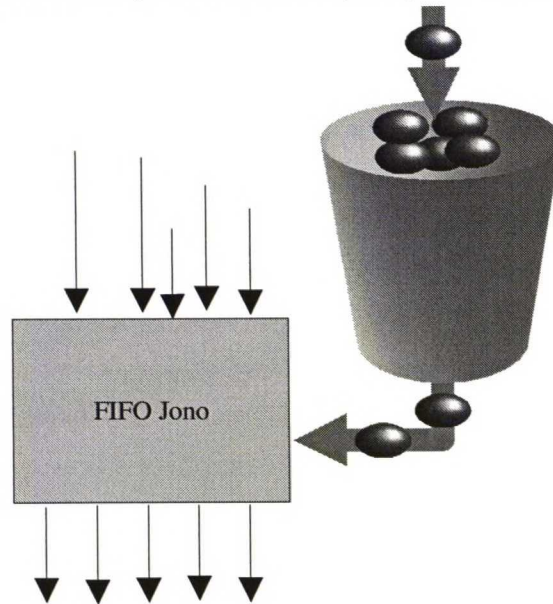
Leaky bucket -mekanismia voidaan verrata sankoon, jonka pohjassa on reikiä. Sankoon kaadettu vesi tulee ulos tasaisina virtoina. Vastaavasti verkossa paketit joutuvat ensin jonoihin, joista niitä lähetetään tasaisena virtana eteenpäin. Mekanismin avulla purskeittainen liikenne voidaan muotoilla tasaiseksi virraksi ja verkkoon lähtevä liikenne tulee ennakoitavaksi ja hallituksi. Menetelmän haittana on sangon täyttyminen. Kun saapuvan liikenteen määrä ylittää sangon läpäisykyvyn ja sanko täyttyy, niin suuri osa saapuvasta liikenteestä valuu sangon reunojen yli. Menetelmä ei myöskään aina käytä verkon resursseja mahdollisimman tehokkaalla tavalla. Kiinteän kokoiset reiät eivät salli liikennepurskeiden hyödyntää vapaata kaistaa, vaikka verkon kuormitus olisi hyvin alhainen.



Kuva 3.1. Liikenteen muokkaaminen leaky bucket -menetelmällä.

Toinen menetelmä on token bucket -mekanismi, jonka olennainen ero edelliseen

mekanismiin on siinä, että liikenne ei kulje lainkaan sangon kautta. Sen sijaan sangossa on vuoromerkkejä, joiden avulla liikennettä ohjataan. Jokaisella virralla on oma sanko, johon merkkejä lisätään tasaisella nopeudella. Kukin vuoromerkki sallii virran lähettää tietyn määrän liikennettä. Mikäli virralla ei ole yhtään vuoromerkkiä, se virta joutuu odottamaan uusia vuoromerkkejä. Toisaalta purskeittainen liikenne voi käyttää kerralla kaikki kertyneet merkit ja menetelmä säilyttää purskeittaisen liikenteen purskeluonteen.



Kuva 3.2. Liikenteen muokkaaminen token bucket -menetelmällä.

Pääsynvalvonta

Pääsynvalvonnan tavoitteena on hallita verkkoon tulevaa liikennettä. Verkossa käytössä oleva pääsypolitiikka määrää, minkälaista ja miten paljon liikennettä verkossa voidaan siirtää. Pääsynvalvonta pitää huolen siitä, että verkon liikenne myös toteuttaa määriteltäviä politiikkaa. Esimerkkinä yksinkertaisesta pääsypolitiikasta on liikenteen suodattaminen IP-otsikon protokollaportin mukaan. Suodattimen läpäisevä liikenne pääsee verkkoon, mutta muu liikenne evätään.

Jonotusalgoritmit

Jonotusalgoritmit ovat reitittimen käyttämiä menetelmiä, joiden avulla reititin päättää, miten saapuvat paketit sijoitetaan jonon ja missä järjestyksessä paketteja lähetetään eteenpäin. Yleisimpiä jonotusalgoritmeja ovat mm. FIFO, Deficit Round Robin (DRR), Class Based Queueing (CBQ), Fair Queueing (FQ), Fluid Fair Queueing (FFQ) ja

Weighted Fair Queueing (WFQ). Kappaleessa 3.3 perehdytään tarkemmin jonotusalgoritmien tarjoamiin vaihtoehtoihin.

Ruuhkautumisen hallinta

Ruuhkautumisen hallinnalla tarkoitetaan mekanismeja, joilla verkko voi pyrkiä torjumaan ruuhkautumista. Mekanismeille on tyypillistä, että ne toimivat yhdessä korkeamman tason protokollien kanssa selvittääkseen ruuhkaa.

Random Early Detection (RED) on Floydin ja Jacobsonin [9] kehittämä algoritmi, joka torjuu ruuhkautumista pudottamalla satunnaisista virroista paketteja jo ennen ruuhkan muodostumista. Algoritmin päätavoitteena on estää kaikkien saman pisteen kautta kulkevien TCP-virtojen synkronoituminen ruuhkatilanteessa. Algoritmi seuraa jonon pituutta, ja kun jono alkaa täyttyä, se valitsee satunnaisesti TCP-virtoja ja pudottaa niiden paketteja, jolloin virtojen lähettäjät voivat vähentää lähetysnopeutta. Mitä täydemmäksi jono täyttyy, sitä useampia virtoja valitaan pudotuksen kohteiksi. Algoritmia voidaan hyödyntää myös muiden virtojen ohjauksessa edellyttäen, että sovellukset osaavat vähentää lähetysnopeutta kadonneitten pakettien perusteella. Bonald [10] on arvioinut algoritmin toimintaa analyttisten mallien avulla ja osoittaa, että RED tasoittaa eri yhteyksien saamaa kaistanleveyttä ja pystyy hallitsemaan keskimääräistä viivettä, mutta johtaa samalla suurempaan viiveenvaihteluun. Lisäksi Bonald toteaa, ettei RED sovellu tasaista bittivirtaa käyttävien sovellusten tarpeisiin.

Pakettien pudottamiseen perustuva ruuhkasta tiedottaminen heikentää väistämättä palvelunlaatua. Parempi keino on käyttää pakettien merkitsemistä. Explicit Congestion Notification (ECN) käyttää kahta IP-otsikon bittiä ilmaisemaan ruuhkaa. Kun reititin huomaa, että verkkoon on muodostumassa ruuhkaa, se asettaa IP-otsikon Congestion Experienced (CE) -bitin läpimenevissä paketeissa. Kun vastaanottaja saa paketin, jonka CE-bitti on päällä, se asettaa kuittauspaketin ECN Echo -bitin päälle. Tämä ilmoittaa lähettäjälle verkon ruuhkautuneen ja lähettäjä voi laskea lähetysnopeutta ennen kuin paketteja joudutaan pudottamaan. [11]

3.3 Lankaverkoissa käytettävät jonotusalgoritmit

Reitittimien käyttämillä jonotusmekanismeilla on keskeinen vaikutus siihen, minkä tasoista palvelua verkko pystyy tarjoamaan [7]. Seuraavassa esitetään lyhyt katsaus muutamiin yleisesti käytettyihin jonotusalgoritmeihin.

3.3.1 First In First Out - jonotus

Internetissä yleisimmin käytetty jonotusalgoritmi on FIFO-algoritmi (First In, First Out). Algoritmi asettaa saapuvat paketit jonon perään ja purkaa jonoa alkupäästä. Jonon täytyttyä saapuvia paketteja pudotetaan kunnes jonossa on jälleen tilaa. Yksinkertaisuutensa takia algoritmi voidaan toteuttaa erittäin tehokkaasti. Algoritmi toimii tehokkaasti niin kauan kun verkossa on tarpeeksi kapasiteettia ja jonossa on riittävästi tilaa liikennepurskeiden vastaanottamiseen. Tilanne muuttuu radikaalisti liikenteen lisääntyessä. Kun jono on täynnä joudutaan useita peräkkäisiä paketteja hylkäämään. Pitkään jatkuvassa ylikuormitustilanteessa verkon tarjoaman palvelun laatu väistämättä heikkenee.

3.3.2 Deficit Round Robin

Deficit Round Robin (DRR) [12] on Naglen [13] Round Robin -algoritmin laajennus. Perinteinen Round Robin -algoritmi käyttää erillistä jonoa jokaista yhteyttä varten ja toimittaa vuorotellen paketteja jokaisesta jonosta. Pakettipohjaisena algoritmi ei kuitenkaan pysty takaamaan eri yhteyksien tasavertaista käsittelyä, mikäli yhteyksien käyttämä pakettikoko vaihtelee. Lisäksi algoritmi skaalautuu huonosti, koska jokaista yhteyttä varten tarvitaan oma jono.

DRR-algoritmi käyttää vajelaskureita, joiden avulla eri kokoisia paketteja sisältäviä pakettivirtoja voidaan käsitellä tasavertaisesti. Sen sijaan, että jokaista yhteyttä varten olisi oma jono, DRR hajauttaa pakettivirrat pieneen määrään jonoja. Jonoja palvellaan antamalla niille vuorotellen tietty määrä lähetyskapasiteettia, jonka puitteissa ne voivat vuorollaan lähettää dataa. Vajelaskurin avulla jonolle hyvitetään seuraavalla kierroksella edellisellä kierroksella käyttämättä jäänyt kapasiteetti. Hajautuksesta johtuen tasavertaisen kohtelun saavuttamiseksi riittää, että jonoja on enemmän kuin aktiivisia yhteyksiä.

3.3.3 Fluid Fair Queuing

Fluid Fair Queuing -algoritmi (FFQ) on suosittu reilun jonotuksen algoritmi, joka käsittelee pakettivirtoja nestevirtausten tavoin. Algoritmi pyrkii jakamaan kapasiteetin virtojen välille käyttäen virroille annettuja painoarvoja. Matemaattisesti FFQ toimii seuraavasti: Mielivaltaisen lyhyessä ajassa t vuon f , jolla on lähetettäviä paketteja, saama kaistanleveys C_f on

$$C_f = C \Delta t \left(\frac{r_f}{\sum_{i \in B_{FFQ}(t)} r_i} \right).$$

Kaavassa C on kanavan kapasiteetti, r_f on pakettivirran f painoarvo ja $B_{FFQ}(t)$ on niiden pakettivirtojen joukko, joilla on paketteja jonossa ajanhetkellä t .

Pakettitason toteutuksia lankaverkkoihin FFQ-algoritmista ovat mm. WFQ [14], WF²Q [15], SCFQ [16] sekä STFQ [17].

Class Based Queueing

Class Based Queueing (CBQ) on yleisnimitys algoritmeille, joissa yksittäisten virtojen sijaan käsitellään useiden virtojen muodostamia joukkoja. Floydin ja Jacobsonin [18] kehittämän algoritmin alkuperäisenä tavoitteena oli jakaa linkki useiden organisaatioiden kesken siten, että ylikuormitustilanteessa kullakin organisaatiolla olisi mahdollisuus käyttää sille varattua osaa linkin kapasiteetista, mutta toisaalta organisaatioiden käyttämättä jääneet resurssit olisivat muiden käytettävissä. Myöhemmin algoritmia on alettu soveltaa myös resurssien jakamiseen eri palveluluokkien välillä [7].

CBQ-algoritmi perustuu liikenteen hierarkkiseen luokitteluun, jossa saapuva liikenne jaetaan ensin luokkiin ja paketit talletetaan lehtiluokkien jonoihin. Sisäsolmuissa olevien luokkien tehtävänä on määrittää ylimääräisen kaistanleveyden jakautuminen lehtiluokille. Jokaiselle luokalle on määritelty osuus linkin kapasiteetista. Algoritmin tavoitteena on taata, että kaikki luokat saavat niille määritellyn osuuden kapasiteetista kohtuullisen ajan sisällä ja jakaa ylimääräinen kapasiteetti reilusti luokkien kesken. Tavoitteiden saavuttamiseksi algoritmi seuraa eri luokkien käyttämien resurssien määrää. Jos luokan liikenne pysyy luokalle määritellyn kapasiteetin rajoissa, luokka saa liikennöidä vapaasti, mutta mikäli se ylittää kiintiönsä, se joutuu joko lainaamaan kapasiteettia tai rajoittamaan liikennettään. CBQ:n tarjoamia palveluita voidaan edelleen parantaa liittämällä luokkiin erilaisia jonotusalgoritmeja ja priorisoimalla osa luokista.

CBQ:n avulla eri palveluluokille voidaan tarjota eri tasoista palvelua, mutta samalla voidaan estää alempien prioriteettiluokkien liikenteen nääntyminen. CBQ:n varjopuolena on sen laskennallinen vaatavuus ja monimutkainen jonokäsittely, jonka seurauksen se ei sovellu nopeisiin verkkoihin.

3.3.4 Weighted Fair Queueing

Weighted Fair Queueing -algoritmin (WFQ) tavoitteena on tarjota virroille ennustettavaa jonotuskäytäntöä, ja pitää samalla huolta siitä, etteivät pienet virrat pääse nälkiintymään. WFQ antaa pienikapasiteettisille virroille etusijan ja antaa isompien virtojen löytää tasapainotilan jäljellä jäävässä kapasiteetissa. Palvelualgoritmin ensisijaisena tavoitteena on tarjota ennustettavia vasteaikoja kaikille virroille. Tähän tavoitteeseen WFQ pääsee sijoittamalla liikennevirtojen paketit eri jonoihin niiden koon perusteella. Tällöin suuret virrat eivät pääse valtaamaan verkon resursseja pienempien virtojen kustannuksella.

WFQ:n ongelmana on jälleen laskennallinen vaativuus. Raskas pakettien järjestely ja jononkäsittely eivät skaalaudu suuriin datamääriin ja nopeisiin verkkoihin. Toinen ongelma WFQ:ssa on rakeisuuden puuttuminen. WFQ ei tarjoa menetelmiä virtojen keskinäisen painotuksen säätämiseen, vaan pyrkii tarjoamaan kaikille virroille tietynlaista tasapainotilaa. Yksi mahdollisuus erilaisten prioriteettien luomiseen on IP-otsakkeen etusija-kentän käyttö. Eri jonojen välinen painotus kuitenkin riippuu algoritmin valmistajakohtaisesta toteutuksesta, joten mekanismin tarjoamat mahdollisuudet ovat rajalliset.

3.4 Palvelunlaadun kehysarkkitehtuurit

3.4.1 Yhdistetyt palvelut

Yhdistetyt palvelut (Integrated Services, IntServ) [8] on IETF:n työryhmän määrittelemä palveluarkkitehtuuri, jonka tavoitteena on mahdollistaa äänen, videon, reaaliaikaisen datan ja perinteisen datan siirto samassa verkossa. Yhdistetyt palvelut määrittelee joukon palveluluokkia, joita tukevat reitittimet pystyvät antamaan datavirroille laatutakuita. Palveluluokat voidaan määritellä virtakohtaisesti sovellusten pyyntöjen mukaan. Pyyntö voidaan välittää verkonhallintarutiineilla tai käyttämällä resurssinvarausprotokollaa. Pyyntö määrittelevät varattavat resurssit sekä välitysmenetelmät, joiden avulla saavutetaan haluttu päästä päähän -laatutaso.

Kun reitittimet tekevät resurssivaroja, niiden on huomioitava linkkikerroksen ominaisuudet sekä valvottava resurssien käyttöä siten, että pyyntö hyväksytään vain, jos resursseja on riittävästi. Kun jokainen reitillä oleva reititin on varannut yhteyden tarvitsemat resurssit, datavirta voi odottaa saavansa pyytämänsä palvelutasoa sillä edellytyksellä, että reitti ei muutu ja datavirta noudattaa pyynnössä määriteltyjä parametreja.

RSVP

Yhdistettyjen palveluiden resurssienvarausprotokollaksi on tarjolla useita vaihtoehtoja. Laajimman kannatuksen on saanut Wroclaskin [19] ehdotus Resource Reservation Protocol:lan (RSVP) [20] käyttämisestä IntServ:n kanssa. Yksinkertaistettuna RSVP toimii siten, että dataa lähettävä sovellus lähettää ensin reitintunnusteluviestin. Tunnusteluviesti välittää vastaanottajalle tiedon siitä, minkälaista dataa lähettäjällä on tarjolla sekä kerää tiedot välissä olevan verkon ominaisuuksista. Vastaanottaja vastaa viestiin lähettämällä varausviestin takaisin samaa reittiä pitkin. Varausviesti varaa reitillä olevilta reitittimiltä tarvittavat resurssit.

Tällä hetkellä RSVP:n kanssa käytettäväksi on määritelty kaksi palvelutasoa paras yritys -palvelun lisäksi. Taattu palvelu (Guaranteed Service) [21] tarjoaa taatun kaistanleveyden ja tiukan kokonaisviiveen ilman jonotuksesta aiheutuvia pakettikatoja. Hallitun kuorman palvelu (Controlled-Load Service) [22] ei sisällä tiukkoja määrällisiä takuita vaan sen tarjoama palvelutaso vastaa kevyesti kuormitetussa verkossa saatavaa paras yritys -palvelua. Toisin kuin paras yritys -palvelussa, hallitun kuorman palvelu takaa saman palvelutason riippumatta verkon kuormituksesta. White on kirjoittanut kattavan johdatuksen RSVP:n ja IntServ:n käyttöön Internetissä [23].

Yhdistetyt palvelut -arkkitehtuuria on kritisoitu sen huonon skaalautuvuuden vuoksi. Virtakohtaisten palvelutakuiden ylläpito vaatii jokaista reititintä pitämään kirjaa jokaisesta yhteydestä, mikä on etenkin runkoverkossa käytännössä mahdotonta. Pienemmässä mittakaavassa yhdistetyt palvelut -arkkitehtuuri on kuitenkin varteenotettava ratkaisu.

3.4.2 Differentiated Services

Yhdistetyt palvelut tarjoaa jokaiselle yhteydelle omaa palvelunlaatua, kun taas palveluluokka-arkkitehtuuri (Differentiated Services) [24] ryhmittelee yhteydet niiden palvelutarpeiden mukaan luokkiin. Luokittelun avulla palvelu skaalautuu myös runkoverkkotasolle.

DiffServ ei käytä päästä-päähän ulottuvaa signaalointiprotokollaa vaan toiminta perustuu palvelutasosopimukseen (Service Level Agreement, SLA). Käyttäjä merkitsee jokaiseen pakettiin haluamansa palvelutason, ja tämän jälkeen verkko kohtelee paketteja tämän tason mukaisesti.

DiffServ-arkkitehtuurin selkeä hyvä puoli on huomattavasti IntServ-arkkitehtuuria parempi skaalautuvuus suuriin verkkoihin. Skaalautuvuuden hintana on luopuminen yksittäisten virtojen hallinnasta. Koska virtoja käsitellään joukkoina, määritellyt palvelutasosopimukset pätevät vain joukoille kokonaisuudessaan. Mikäli jonkun luokan liikenne ylittää sille määritellyn kapasiteetin, yksittäisten virtojen saamaa palvelutasoa ei enää kyetä takaamaan. Runkoverkkotasolla, jossa on tuhansia yhtäaikaista virtoja, todennäköisyys yksittäisen virran epäoikeudenmukaisesta kohtelusta on pieni. Pienemmissä verkoissa tämä todennäköisyys kuitenkin kasvaa, ja algoritmin toimivuus pienissä verkoissa on kyseenalaista.

IP-paketin otsikossa sijaitseva TOS (Type of Service) kenttä on määritelty tukemaan pakettien luokittelua [24]. Kun paketti saapuu DiffServ-verkkoon, paketin palvelukoodi asetetaan vastaamaan paketin tarvitsemaa palvelutasoa. Palvelukoodi määrää, miten pakettia kohdellaan sen kulkiessa verkon sisällä. Verkon rajalla reitittimet tarkastavat, että merkityt paketit noudattavat kullekin palvelutasolle määriteltyjä sääntöjä. Palveluluokat on toteutettu kuvaamalla IP-otsikon sisältämä palvelukoodi hyppykohtaiseksi käytökseksi (per-hop behaviour, PHB) jokaisessa matkan varrella olevassa verkko-elementissä. Eri tarkoituksiin on määritelty erilaisia PHB:ja. Esimerkiksi taattu toimitus (AF PHB) tarjoaa asiakkaalle tietyn minimiläpimenon ruuhkaisessakin verkossa [25].

3.4.3 Yhdistelmäarkkitehtuuri

IntServ- ja DiffServ-arkkitehtuurien yhdistelmäarkkitehtuurilla voidaan tavoitella molempien arkkitehtuurien etuja. Paikallisverkkojen pienet liikennemäärät sallivat IntServ-pohjaisten ratkaisujen käytön ja mahdollistavat DiffServ-luokkia tarkemman yhteyskohtaisen palvelunlaadun takaamisen. Runkoverkkojen suuri kapasiteetti ja käyttäjämäärät taas ovat DiffServ-arkkitehtuurin ominta aluetta. Yhdistelmäarkkitehtuurin haasteena onkin näiden kahden arkkitehtuurin tehokas yhteensovittaminen.

Mamais et al. [26] esittelemässä yhdistelmäarkkitehtuurissa paikallisverkoissa sijaitsevat käyttäjät lähettävät varauspyynnöt IntServ-mallin mukaisesti RSVP:tä käyttäen. Runkoverkko perustuu kuitenkin DiffServ-arkkitehtuuriin. Runkoverkon rajalla sijaitseva reititin on yhdistelmä RSVP- ja DiffServ-reititintä. Tämä rajareititin hallitsee paikallisten resurssien käyttöä, muuntaa ulospäin lähtevät varauspyynnöt niitä vastaaviksi DiffServ-palveluluokkapyynnöiksi sekä lähettää RSVP-viestin kaistanvälittäjälle (*bandwidth broker*). Kaistanvälittäjät hallitsevat DiffServ-verkon resursseja ja päästä päähän

-palvelunlaatua. DiffServ-verkossa liikkuville RSVP-viesteille on varattu oma palveluluokka, joka takaa signaloinnin tehokkaan toiminnan.

3.5 Palvelunlaatu langattomissa verkoissa

Langaton verkko on palvelunlaadun kannalta erityisen ongelmallinen, koska käyttäjien liikkuvuus aiheuttaa verkosta riippumattoman epävarmuustekijän liikennöinnin onnistumiselle. Lankaverkossa palvelunlaadun takaaminen toimii staattisesti. Kun reitti lähettäjän ja vastaanottajan välillä on kerran muodostettu, se säilyy samana koko yhteyden ajan. Langattomassa verkossa reittiä joudutaan päivittämään aina kun käyttäjä siirtyy uuden yhteyspisteen alueelle. Reitin päivityksen onnistuminen riippuu mm. siitä, onko uusilla reitittimillä tarpeeksi resursseja siirtyvää käyttäjää varten. Toisaalta käyttäjä voi siirtyä langattoman verkon kantoalueen ulkopuolelle, joten absoluuttisten takeiden antaminen palvelunlaadusta on joka tapauksessa mahdotonta.

Langattomaan verkkoon voidaan kuitenkin kehittää järjestelmiä, joilla voidaan tarjota hyväksyttävän tasoista palvelunlaatua liikkuville käyttäjille. Yksinkertaisimmillaan langatonta käyttäjää kohdellaan staattisena, eikä liikkuvuutta tueta palvelunlaadun kanssa. Verkko voi myös pyrkiä ennustamaan käyttäjien liikkumista ja varata viereisistä soluista etukäteen kapasiteettia. Tällöin ongelmiksi muodostuvat liikkumisen ennakointi ja turhien varausten aiheuttama kapasiteetin hukkaus.

3.5.1 Algoritmien soveltaminen langattomiin verkkoihin

Lankaverkkoihin suunniteltuja algoritmeja ei voida suoraan hyödyntää langattomissa verkoissa, koska ne olettavat kaikkien pakettivirtojen olevan tiedonsiirron kannalta samanarvoisia. Langattomissa verkoissa tilanne on kuitenkin toinen. Jokaisella päätelaitteella on erilainen siirtotie, jonka ominaisuudet ja kapasiteetti riippuvat mm. päätelaitteen sijainnista. Tällöin voi syntyä tilanteita, joissa joidenkin pakettivirtojen lähettäminen onnistuu, mutta samanaikaisesti toisten pakettivirtojen lähetys ei onnistu lainkaan. Lankaverkkoihin suunnitellut algoritmit olettavat kanavan olevan virheetön ja yrittävät lähettää paketin, vaikka vastaanottaja olisi sillä hetkellä tavoittamattomissa. Linkkikerroksen uudelleenlähetysalgoritmi kuluttaa tällöin resursseja hukkaan yrittäessään väkisin lähettää paketteja, joiden toimitus ei sillä hetkellä onnistu.

3.5.2 Kanavan tilan seuranta

Langattoman verkon virheille on tyypillistä tietty purskeittaisuus, joka johtuu mm. muiden lähettimien aiheuttamista virheistä. Bhagwat et al. [27] esittelemä menetelmä, Channel State Dependent Packet Scheduling (CSDPS), pyrkii parantamaan langattoman verkon käyttöastetta hyödyntämällä tietoa kanavan tilasta. Käyttöasteen lisääntyminen perustuu siihen, että kun tukiasema huomaa jonkun kanavan olevan 'huono', se ei yritä väkisin lähettää päätelaitteelle paketteja vaan lähettää välillä muille asemilla palaten seuraavalla lähetyskierröksellä jälleen kokeilemaan linkin tilaa. Vaikka CSDPS yrittääkin maksimoida linkin käyttöasteen, niin se ei kuitenkaan tarjoa minkäänlaisia mekanismeja kaistanleveyden jakamiseksi eri yhteyksien välille eikä näin ollen yksistään sovellu palvelunlaadun tarpeisiin.

3.5.3 Luokkaperustainen jonotus kanavan tilan seurannalla

Fragoulis, Sivaramanin ja Srivastavan [28] kehittämä langattoman linkin jakomenetelmä yhdistää CBQ-algoritmin langattoman laajennuksen kanavan tilan seurantaan. CBQ-laajennus sisältää kaksi parannusta alkuperäiseen algoritmiin verrattuna. Ensinnäkin se pyrkii takaamaan jokaiselle luokalle sille määritellyn osuuden tehollisesta kaistanleveydestä, joka on kulloinkin läpimenevän kokonaisliikenteen määrä. Toiseksi kapasiteettinsa ylittäneillä luokilla on mahdollisuus lähettää paketteja, mikäli niillä on hyvä kanava ja kaikilla alikapasiteetista kärsivillä luokilla on huono kanava. Fragoulis et al. suorittivat simulointeja, jotka osoittivat, että CBQ+CSDPS pystyy jakamaan langattoman verkon kapasiteetin reilusti eri luokkien välille. Erityisesti huonoilla linkeillä menetelmän ylivoimaisuus verrattuna CBQ:hun, CSDPS:ään ja pelkkään FIFO-ajoitukseen korostui.

3.5.4 Reilu jonotus langattomissa verkoissa

Yksi tapa parantaa langattoman verkon tehokkuutta on jonotuksen toteuttaminen MAC-tasolla. MAC-tasolle toimiva jonotus pystyy hyödyntämään kanavan tilatietoa tehokkaammin kuin IEEE 802.11 -standardin päälle rakennettu järjestelmä. Lun, Bharghavan ja Srikantin [29] esittävät, että FFQ ei sovellu reilun jonotuksen perustaksi langattomissa verkoissa. Heidän esittelemänsä jonotusmalli, Idealized Wireless Fair Queueing -algoritmi (IWFQ), on FFQ:n muunnos, joka huomioi sijainnista riippuvien virhepurskeiden vaikutuksen. Ideaalisen mallin lisäksi he esittelevät MAC-protokollan, joka tarjoaa pakettitason toteutuksen IWFQ -algoritmista. Toteutus ennakoii kanavavirheitä edellisen lähetyksen perusteella ja järjestää lähetettävien pakettien

järjestyksen siten, että purskeittaisten virheiden vaikutus saadaan minimoitua. Tällöin hyvillä kanavilla liikennöivät yhteydet saavat lähettää enemmän kuin huonoja kanavia käyttävät yhteydet. Tämä onkin tehokkuuden kannalta välttämätöntä. Reiluuden takaamiseksi jälkeenjääneillä jonoilla on mahdollisuus kiertää kiinni välimatka enemmän lähettäneisiin jonoihin.

Protokollan käyttöä rajoittaa sen toteutus MAC-kerroksessa. Koska se ei ole yhteensopiva IEEE 802.11 -standardiin pohjautuvien langattomien verkkojen kanssa, suuri osa nykyisistä langattomista verkoista jää käyttöalueen ulkopuolelle.

4 Multimedia Internetissä

Tässä kappaleessa esitellään yleisimmät multimediasovellukset ja tarkastellaan niiden vaatimuksia palvelunlaadulle. Käsittely aloitetaan tarkastelemalla, minkälaisista elementeistä erilaiset multimediasovellukset koostuvat ja minkälaisia yleisiä vaatimuksia erilaisilla multimedian sovellusalueilla on. Seuraavaksi tarkastellaan mediaelementtien siirtämistä ja erilaisten mediaelementtien tiedonsiirtotarpeita. Lopuksi tarkastellaan eri elementtien yhteisvaikutusta tiedonsiirron kannalta ja palvelunlaadun vaikutusta esityksen laatuun.

4.1 Multimediasovelluksen rakenne

Valtaosa multimediasovelluksista koostuu erillisistä kohtauksista, joiden välillä käyttäjä pystyy navigoimaan. Esimerkiksi tilausvideo (*video-on-demand, VoD*) voi sisältää alussa videon valinta osuuden, joka esittelee tarjolla olevia videoita käyttäen grafiikka, kuvia ja tekstiä. Sen jälkeen, kun video on valittu, videosta tulee sovellusta hallitseva elementti, joka ei juurikaan tarvitse muita elementtejä esityksen tueksi.

Kohtaukset puolestaan rakentuvat joukosta mediaelementtejä ja vuorovaikutteisia komponentteja. Mediaelementit sisältävät varsinaisen esitettävän tiedon audion, videon, grafiikan, kuvien ja tekstin muodossa. Vuorovaikutteiset komponentit, kuten hyperlinkit ja ohjainnäppäimet, antavat käyttäjälle mahdollisuuden hallita multimediaesityksen kulkua. Komponentit ja mediaelementit ovat sovelluksen ulospäin näkyvä osa, mutta sen lisäksi tarvitaan ohjaustietoa, joka määrittelee eri elementtien ja komponenttien sijainnin esityksessä. Tähän tehtävään on kehitetty useita standardeja, joista tunnetuimpia ovat HTML, MHEG ja SMIL.

4.1.1 Multimedian kuvauskielet

HTML

Hyper Text Markup Language (HTML) [30] on erityisesti WWW-julkaisuun suunniteltu sivunkuvauskieli, jonka avulla voidaan mm.

- Julkaista on-line dokumentteja, jotka sisältävät mm. otsikoita, tekstiä ja kuvia.
- Hakea tietoa hyperlinkkien kautta nappia painamalla.
- Käyttää lomakkeiden avulla etäpalveluita, kuten tiedonhakuja tai varausjärjestelmiä.
- Liittää mm. video- ja ääninäytteitä suoraan dokumentteihin.

HTML perustuu SGML:ään ja sen avulla tekstidokumenttiin voidaan liittää tietoa dokumentin rakenteesta ja ulkonäöstä. Lisäksi dokumenttiin voidaan linkittää erilaisia elementtejä, kuten kuvia ja ääntä. HTML soveltuu kuitenkin huonosti multimediamateriaalin esittämiseen, koska se ei tarjoa mahdollisuutta tarkasti asemoida eri elementtejä, eikä mahdollista eri mediaelementtien välistä synkronointia.

WWW-palvelujen tarjoajat ovat pyrkineet laajentamaan HTML:n mahdollisuuksia lisäämällä siihen dynaamisia elementtejä, joilla HTML-pohjaisten sivujen avulla voidaan esittää entistä rikkaampaa multimediasisältöä. Tästä laajennetusta HTML:stä käytetään termiä Dynaaminen HTML (DHTML). Se tarjoaa mahdollisuuden elementtien tarkkaan asemointiin ja tyylin määrittelyyn tyylisivujen (CSS) avulla sekä mahdollistaa HTML:ää laajemmat mahdollisuudet muokata sivua dynaamisesti skriptikielten avulla. [31]

HTML-sivu itsessään sisältää ainoastaan sivun tekstiosuuden sekä tiedot muista elementeistä, joten sivun koko vaihtelee tyypillisesti vajaasta kilotavusta kymmeneen kilotavuuihin. Sivuuun liittyvät elementit, kuten kuvat, ladataan omista tiedostoistaan, ja ne voivat nostaa koko sivun esittämiseksi tarvittavan tiedonsiirtomäärän satoihin kilotavuuihin. Useimmat HTML-sivut ovat staattisia, jolloin koko sivu ladataan kerralla selaimen muistiin esittämistä varten. Usein selaimet esittävät sivun rungon ennen sivulle linkitettyjen kuvien yms. elementtien lataamista, mutta käytännössä WWW-sivujen käyttö edellyttää yleensä kaiken sivuun kuuluvan datan siirtämistä.

MHEG

MHEG on erityisesti multimediaelementtien yhdistämiseen suunniteltu standardiryhmä, jonka avulla voidaan hallita multimedian ja hypermedian esittämistä. MHEG tarjoaa mm. mahdollisuuden määritellä, miten videota ja audiota esitetään ja miten käyttäjä voi vaikuttaa esityksen kulkuun. Toisin kuin merkkaukieleen pohjautuva HTML, MHEG käyttää skripteihin pohjautuvaa mekanismia multimediaesityksen rakenteen kuvaamiseen ja tarjoaa mahdollisuuden mm. eri mediaelementtien synkronointiin. MHEG hyödyntää oliomallia mediaelementtien, vuorovaikutuksen ja toimintojen kuvaamiseen. [1]

MHEG tarjoaa laitteistosta riippumattoman standardin multimediaesitysten luomiseen ja sitä voidaan hyödyntää erilaisilla alustoilla, kuten vuorovaikutteisissa etäopetuksessa, tilausvideossa, interaktiivisessa TV:ssä ja multimediakioskeissa. Erityisesti vuorovaikutteisen multimedian jakeluun kehitetty MHEG-5 -standardi tarjoaa asiakas/palvelin arkkiteh-

tuuriin perustuvan ratkaisun, jolla MHEG sovelluksia voidaan ajaa verkon yli erilaisilla päätelaitteilla. Internetissä MHEG on kuitenkin jäänyt yksinkertaisempien standardien, kuten HTML:n ja SMIL:in, varjoon. [1]

SMIL

SMIL (Synchronized Multimedia Integration Language) on W3C:n kehittämä standardi multimedian esittämiseksi WWW:ssä. SMIL perustuu XML-pohjaiseen merkintäkieleen ja muistuttaa läheisesti HTML:ää. HTML:ään verrattuna SMIL tarjoaa paremmat työkalut multimedian käsittelyyn: esityksen layout voidaan määritellä pikselin tarkkuudella, eri mediaelementeille voidaan määritellä vaihtoehtoisia esitysmuotoja ja median toisto voidaan ajoittaa. SMIL:n sovelluskohteita ovat mm. vuorovaikutteinen video, tilausvideo, online-opetus ja multimedia tietopalvelut. [32]

SMIL-esitys koostuu mediaa sisältävistä alueista. Kullekin alueelle on määritelty mediatiedosto, joka sisältää alueella esitettävän mediadatan. Median esitys voi tapahtua peräkkäin tai rinnakkain ja esitykseen voidaan liittää grafiikkaa, tekstiä ja hyperlinkkejä. Esityksessä voidaan hyödyntää lukuisia erilaisia SMIL:n tukemia mediatyyppejä, kuten virtaavaa videota ja audiota, kuvia, animaatiota, grafiikkaa ja tekstiä, ja yhdistellä niistä vuorovaikutteinen multimediaesitys. Vaihtoehtoisten mediaelementtien avulla käyttäjille voidaan tarjota eri versioita mediasta riippuen mm. käyttäjän kielestä, tietoliikennekapasiteetista ja mieltymyksistä. [32]

4.2 Multimedian siirtostandardit

Erilaisten mediaelementtien tiedonsiirtovaatimukset eroavat huomattavasti. Grafiikka, kuvat ja teksti siirretään tyypillisesti kokonaisina elementteinä, jolloin niiden sujuva esittäminen vaatii hetkellisesti kohtuullisen suurta tiedonsiirtokapasiteettia. Elementtien siirryttyä kaistanleveyttä ei kuitenkaan enää tarvita. Audion ja videon siirrosta dataa pyritään yleensä esittämään jatkuvasti sitä mukaan kun dataa siirtyy. Vallitsevana teknologiana Internetissä on virtaava data, joka vaatii jatkuvaa häiriötöntä tiedonsiirtokapasiteettia. Toinen siirtovaihtoehto on progressiivinen siirto, joka vastaa enemmän perinteistä tiedonsiirtoa siinä mielessä, että jatkuva tiedonsiirtokapasiteetti ei ole ehdoton välttämättömyys. Progressiivisessa siirrosta data siirretään sovelluksen kannalta yhtenä tiedostona, mutta sen esitys aloitetaan jo siirron aikana. Tällöinkin keskimääräisen kapasiteetin on täytettävä tietyt minimivaatimukset, mutta hetkelliset kapasiteetin vaihtelut eivät haittaa esitystä. Multimedian siirtoon tarvittava kapasiteetti riippuu hyvin

pitkälle audion ja videon laadusta sekä käytetystä tekniikasta. Laadukkaan audion siirto vaatii noin 128 kbps:n kaistanleveyden ja videoon tarvitaan lähes 1 Mbps:n kaistanleveyttä. Laatuvaatimuksista tingittäessä tullaan toimeen audion siirrossa muutamilla kilobiteillä sekunnissa ja videon siirrossa muutamilla sadoilla kilobiteillä sekunnissa. Tällöin esityksen laatu kuitenkin jää usein liian heikoksi pidempiaikaista seuraamista varten. Seuraavissa kappaleissa tarkastellaan tarkemmin erilaisten mediaelementtien esitystä erityisesti Internetissä.

4.2.1 Teksti, grafiikka ja kuvat

Teksti, grafiikka ja kuvat muodostavat perinteisen multim mediasovelluksen ytimen. Erityisesti teksti ja grafiikka ovat tiedonsiirron kannalta kevyitä medioita, jotka voidaan siirtää Internetissä teksti- tai HTML-tiedostoina. Grafiikan ja tekstin osuus esimerkiksi HTML-sivuilla jää yleensä muutamiin kilotavuihin.

Kuvat ovat tällä hetkellä yksi multim mediasovellusten tärkeimpiä komponentteja, joiden avulla sovelluksen visuaalinen ulkoasu muodostetaan. Kuvat vievät huomattavasti enemmän tilaa kuin teksti tai grafiikka, joten niiden siirrossa hyödynnetään pakkausta. Kaksi vallitsevaa kuvien siirtomuotoa Internetissä ovat GIF ja JPEG. Näiden avulla useimpien kuvien koko voidaan pudottaa muutamiin kilotavuihin, [1]

GIF-kuvat käyttävät hävittämätöntä pakkausmenetelmää, joka soveltuu erityisesti grafiikkaa sisältävien kuvien pakkaamiseen [1]. JPEG puolestaan on suunniteltu jatkuvasävyisten kuvien pakkaamiseen. JPEG perustuu häviölliseen pakkausalgoritmiin, joka jakaa kuvan 8x8 pikselin kuvalohkoihin. Kuvalohkot muunnetaan taajuusavaruuteen, jossa ihmisen näön kannalta epäoleellista informaatiota vähennetään kvantisoinnilla. Lopuksi kvantisoitu kuvadata pakataan vielä Huffman-koodauksella. Menetelmällä voidaan saavuttaa jopa pakkaussuhde 20:1 ilman näkyvää vaikutusta kuvaan, ja jopa pakkaussuhde 100:1 on käyttökelpoinen esikatselu- ja luettelointitarkoituksissa. Tarkkojen ja hyvälaatuisten JPEG-kuvien koko jää kuitenkin satoihin kilotavuihin, ja yleensä näin isot kuvat ovat erillisten linkkien takana, joten niiden lataaminen on käyttäjän valittavissa. [33]

Näyttävillä WWW-sivuilla kuvien yhteenlaskettu koko nousee helposti satoihin kilotavuihin. Mikäli käyttäjä haluaa kaikkien sivulla olevien kuvien latautuvan kohtuullisessa ajassa, esimerkiksi viidessä sekunnissa, tarvitaan jo kuvien siirtoon 50-100 kbps:n kaistanleveys sivun latautumisen ajaksi. Sivujen latautumista voidaan nopeuttaa mm.

kuvien uusiokäytöllä, jolloin useat sivut hyödyntävät samoja kuvia, sekä kuvien progressiivisella pakkauksella. Progressiivisessa pakkauksessa kuva koodataan siten, että siitä voidaan ensin näyttää karkea versio, jota päivitetään tarkemmaksi sitä mukaa kun kuvadataa saapuu perille. Tällöin käyttäjän ei tarvitse odottaa koko kuvan latautumista. [1]

4.2.2 Video

Video on tiedonsiirron kannalta erittäin raskas media. Digitalisoitu pakkaamaton TV-kuva tarvitsee yli 100 Mbps:n kaistanleveyden ja studiotasoiset digitaaliset videosaatimet voivat viedä yli 200 Mbps. Perusmenetelmät, joilla kaistanleveyttä voidaan pienentää, ovat alinäytteistys, kuvadatan pakkaaminen sekä kuvanopeuden lasku. Ihmisen silmä havaitsee kirkkauserot värierot tarkemmin, joten digitaalisessa videossa yleensä tingitään krominanssi-informaation tarkkuudesta. Krominanssin vaakasuuntaisen näytteistystarkkuuden puolittaminen (4:2:2) tuo 33% säästön verrattuna alkuperäiseen signaaliin. Edelleen puolittamalla myös pystysuuntainen tarkkuus (4:2:0) saavutetaan 50% säästö alkuperäiseen signaaliin verrattuna ilman suurempia vaikutuksia kuvanlaatuun. [1]

Korkean pakkaustiheyden saavuttamiseksi videonpakkauksessa hyödynnetään videossa esiintyvää tila- ja aikaredundanssia. Tilaredundanssin minimoimiseksi yksittäiset kuvat voidaan pakata JPEG-tyyppisellä pakkausalgoritmilla. Pakatun kuvan tarkkuutta ja kokoa voidaan säätää muuttamalla kvantisoinnin rakeisuutta. Videokuvia, jotka hyödyntävät ainoastaan tilaredundanssia, kutsutaan perus- tai I-kuviksi (*Intraframe coding*). Videossa peräkkäiset kuvat ovat usein lähes samanlaisia. Tästä syntyvää aikaredundanssia voidaan hyödyntää koodaamalla kuvista ainoastaan muuttuneet osat. Kuvia, jotka on koodattu edellisten kuvien avulla, kutsutaan erotus- tai P-kuviksi. [1]

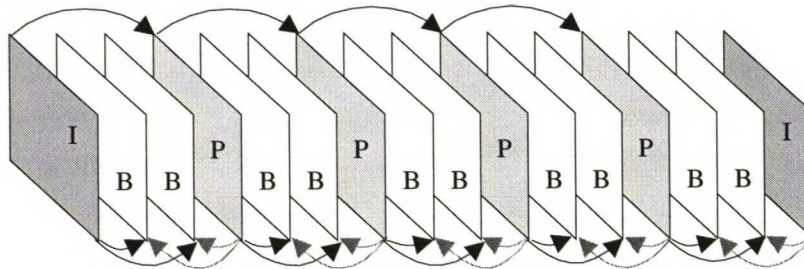
Yleisimmät Internetissä käytetyt videonpakkausmenetelmät ovat Motion Picture Experts Group'in MPEG-standardit, Applen Quicktime, Real Networksin RealVideo sekä Microsoftin Windows Media Player.

MPEG-1

MPEG-standardiperheeseen kuuluvat videonkoodausstandardit ovat MPEG-1, MPEG-2 ja MPEG-4. MPEG-1-standardin [34] tavoitteena oli digitaalisen videon tallentaminen digitaaliselle medialle 1.5Mbps bittinopeudella, joka mahdollistaa VHS-tasoisien videon tallettamisen esim. CD-levylle. Standardimuotoisen MPEG-1 videon resoluutio (CIF) on 352x240x30 kuvaa/s (NTSC) tai 352x288x25 kuvaa/s (PAL). Standardi ei rajoita

resoluutiota tai kuvanopeutta, mutta suuret poikkeamat standardiarvoista heikentävät videon laatua huomattavasti. Internetissä joudutaan tyytymään huomattavasti alempiin nopeuksiin ja alempaan videonlaatuun. Usein käytetään neljännesresoluutiota (QCIF) 176x120 10-15 kuvaa/s kuvanopeuksilla. Lopputuloksen laatu jää melko heikoksi, mutta se riittää monien internet-sovellusten tarpeisiin. Julkisenä standardina MPEG-1 on laajalti käytössä ja soveltuu siten hyvin videon levitykseen.

MPEG-virta koostuu kuvaryhmistä (Group of Pictures, GOP), jotka koostuvat edelleen yksittäisistä kuvista. Kuvia on kolme päätyyppiä: I-kuvia, P-kuvia ja B-kuvia. I-kuvat ovat sisäisesti koodattuja kuvia, joiden pakkausalgoritmi on JPEG-pakkauksen kaltainen. P-kuvat ovat ennustavasti koodattuja kuvia, jotka hyödyntävät edellisen P/I-kuvan kuvaelementtejä. B-kuvat hyödyntävät sekä edellisen että seuraavan P/I-kuvan kuvaelementtejä. Ennustavasti koodattujen kuvien muodostuksessa naapurikuvan kuvaelementeistä muodostetaan mahdollisimman yhdenmukainen kuva, jonka jälkeen lasketaan erotus koostetusta kuvasta ja alkuperäisestä kuvasta. Tämä erotus koodataan JPEG-tyyppisellä algoritmilla ja lähetetään yhdessä kuvaelementtien liikevektoreiden kanssa. Kuvaryhmän tyypillinen rakenne on esitetty kuvassa 4.1.



Kuva 4.1. MPEG-kuvien riippuvuus toisista kuvista.

Kuvien ennustava koodaus nostaa pakkaustiheyttä, mutta samalla tekee virran herkemäksi virheille, koska kuvien virheet periytyvät seuraaviin kuviin. Esimerkiksi 30 kuvaa/s kuvanopeudella yksittäinen virhe I/P-kuvassa voi periytyä jopa 14 muuhun kuvaan ja olla näkyvissä maksimissaan 0,5 sekuntia. Kokonaisen I-kuvan menetyksestä seuraa aina puolen sekunnin katko esitykseen, koska koko kuvaryhmä menetetään.

Boyce ja Gaglianella [35] tutkivat, miten pakettikato Internetissä vaikuttaa MPEG-1 koodattuun videoon. Tutkimuskohteena oli UDP:n päällä siirrettävä virtaava MPEG-1 -video. Tulosten perusteella jo kolmen prosentin pakettikato voi aiheuttaa virheen 30 prosenttiin kuvista. Tämä osoittaa selvästi tarpeen käyttää virheiden piilotus- ja/tai korjausmenetelmiä, kun MPEG-videota lähetetään Internetissä.

MPEG-2

MPEG-2 -standardi [36] on kehitetty TV-tasaisen digitaalisen videon talletukseen, työstöön ja lähetykseen. Standardi perustuu MPEG-1:een, mutta on sovitettu suuremmille bittinopeuksille. Edeltäjänsä verrattuna se tarjoaa huomattavasti paremman kuvanlaadun. Normaali TV-tasoinen MPEG-2 -lähetykseen, jonka tarkkuus on 720 x 576x 30 kuvaa/s, kuluttaa noin 4,5 Mbps kaistaa. MPEG-2 tarjoaa myös laajan valikoiman muita tarkkuuksia välillä 352x288x30 kuvaa/s - 1920x1152x60 kuvaa/s, jotka vastaavat bittinopeuksia 2 Mbps:sta aina 100 Mbps:iin asti. MPEG-2:sta käytetään mm. DVD-elokuvissa sekä digitaalisessa televisiossa. Internetissä ja eritoten langattomissa verkoissa näin suuri kaistan tarve on kohtuutonta, joten MPEG-2:n käyttö rajoittuikin erittäin hyvää laatua edellyttäviin sovelluksiin.

MPEG-4

MPEG-perheen tuorein tulokas on MPEG-4, joka yhdistää videon ja multimedian samaan standardiin. Standardi määrittelee menetelmät, joiden avulla multimedia-aineistoa voidaan luoda ja siirtää sekä määrittelee, miten käyttäjät voivat vaikuttaa esitykseen. MPEG-4 esittää multimedian joukkona audiovisuaalisia objekteja (AVO), joista muodostuu esityksen audiovisuaalinen maisema. Erilaisia objekteja ovat mm. teksti-, grafiikka, animaatio- ja video-objektit. Skaalautuvien esitysmuotojen avulla voidaan erilaisia päätelaitteita käyttäviä asiakkaita palvella kunkin päätelaitteen ominaisuuksien mukaan. [1]

MPEG-4 -standardiryhmän videokoodikit tarjoavat laajan valikoiman erilaisia bittinopeuksia. Hyvin alhaisen bittinopeuden (VLBV) koodaus on suunniteltu mm. langattomien sovellusten käyttöön ja tarjoaa 5-64 kbps bittinopeudet. MPEG-4 pystyy koodaamaan vähän liikettä sisältävää videota, kuten videopuhelua, hyvin tehokkaasti käyttämällä pitkiä ennustavien kuvien jaksoja. Esimerkiksi SQCIF resoluution video (160x120) voidaan koodata 12-28 kbps:n bittinopeudella käyttäen 120 kuvan ennustavasti koodattuja jaksoja kuvanopeuksilla 5-15 kuvaa/s. TV- ja elokuvasovelluksia varten MPEG-4 tarjoaa korkeamman bittinopeuden videokoodauksen (HBV), joka sisältää bittinopeuksia 4 Mbps:iin asti.

MPEG-4 tukee tiedonsiirtoa erilaisissa verkkoympäristöissä ja sisältää mekanismeja, joilla multimediaa voidaan siirtää luotettavasti virheille herkillä kanavilla. Alhaisen bittinopeuden ansiosta MPEG-4 sietää bittivirheitä paremmin kuin MPEG-2. Esimerkiksi

kun verrataan MPEG-4:sta ja MPEG-2:sta kanavalla, jonka BER on 10^{-6} , 3 Mbps:sa koodatussa MPEG-2 videossa esiintyy 180 virhettä minuutissa, kun taas MPEG-4 -virrassa, joka on koodattu 50 kbps:ssa, vastaava virhemäärä on vain 3 virhettä minuutissa. Jos videoiden kuvanopeudet ovat 30 kuvaa/s ja 10 kuvaa/s MPEG-2:lle ja MPEG-4:lle, niin keskimäärin MPEG-2 -videossa on 20 kertaa enemmän virheitä kuin MPEG-4 -videossa. Virheet MPEG-4 -videossa voivat kuitenkin olla pahemman näköisiä ja niiden vaikutus olla havaittavissa pidempään kuin MPEG-2 -videossa. Tämä johtuu MPEG-4:n suuremmasta pakkaustiheydestä. Mikäli molemmissa videoissa käytetään yhtä pitkiä ennustavasti koodattuja kuvasarjoja, virheet MPEG-4 -videossa säilyvät 4 kertaa pidempää näkyvissä kuin MPEG-2 -videossa. Alhaisilla bittinopeuksilla MPEG-4 voi hyödyntää MPEG-2:sta pidempiä ennustavasti koodattuja kuvasarjoja, jolloin ennustavasti koodattujen sarjojen pituus voi olla 5-10 -kertainen verrattuna MPEG-2:een. Tällöin virheiden leviäminen lisääntyy huomattavasti. Monet MPEG-4 -sovellukset eivät kuitenkaan tarvitse MPEG-2:n tasoista videolaatua ja monissa ympäristöissä MPEG-4:n laatu on riittävä jopa ilman MPEG-4:n virheistäpalautumisominaisuuksia. [37]

MPEG-4:n virheistäoipumismenetelmiin kuuluvat virheiden tunnistus ja paikannus, virheellisen datan palautus sekä virheiden visuaalinen piilotus. Virheiden paikantamiseksi MPEG-4 tarjoaa ns. pakettivideotilan, jossa videovirta koostuu itsenäisistä videopaketeista. Videopakettien avulla virhe voidaan eristää siihen videopakettiin, jossa se esiintyi. Virheellinen data voidaan yrittää palauttaa ympäröivän datan avulla. Datan osituksella voidaan edelleen parantaa virheen paikannusta. Datan osituksessa kuvan liike- ja otsikkotiedot on erotettu itse kuvadatasta. Jos kuvadatan siirrossa tulee virhe, voidaan virheestä aiheutuvia visuaalisia virheitä korjata edellisen kuvan elementtien ja liikevektoreiden avulla. [38]

Toinen menetelmä, jolla virheellisen datan vaikutusta pienennetään, on käännettävien vaihtuvamittaisten koodien käyttö. Mikäli eteenpäin tulkittaessa kohdataan virheellinen koodisana, virta voidaan avata taaksepäin lähtien seuraavan videopaketin rajalta. Lisäksi MPEG-4 voi vähentää virheiden leviämistä käyttämällä kuvan sisäistä virkistämistä. Tämä tarkoittaa tiettyjen kuvan osien koodaamista I-kuvina. Tällä menetelmällä voidaan esimerkiksi paljon liikettä sisältäviä kuvan osia päivittää useammin kuin vähän liikettä sisältäviä osia. [38]

Perustason virheistöipumismekanismien avulla MPEG-4 -video (QCIF, 120kbps) säilyy hyväksyttävänlaatuisena, kun siirtotien virhetiheys on alle 10^{-5} . MPEG-4:ään sisältyvien virheensietomekanismien avulla videon laatua voidaan parantaa alle 10%:n lisäkustannuksilla, jolloin on mahdollista saavuttaa hyväksyttävän laatuinen videolaatu vielä kun pakettikato on 10%. [37]

Quicktime

Quicktime on Applen kehittämä standardi, joka mahdollistaa useista eri medioista koostuvan esityksen yhdistämisen yhdeksi videoksi. Itse asiassa Quicktime:a voidaan pitää multimedian esitysmuotona, jonka yksi keskeinen sovellus on nimenomaan videoiden välitys. Quicktime:n rakenne on pitkälti sama kuin MPEG-4:n ja MPEG-4-tiedostojen rakenne pohjautuukin Quicktime:n käyttämään rakenteeseen. Quicktime tukee sekä virtaavaa videota RTP:n ja RTSP:n päällä että progressiivista videon siirtoa, jossa video siirretään FTP:n tai HTTP:n päällä. Progressiivisen videon keskeisin etu on mahdollisuus siirtää videota myös hitaiden ja häiriöherkkien kanavien ylitse, koska FTP ja HTTP pitävät huolta siitä, että kaikki videopakettit pääsevät perille siirtovirheistä huolimatta. Virtaavan videon etuja ovat puolestaan mm. mahdollisuus suoriin lähetyksiin sekä pienempi kaistanleveys. [39]

Suosituin Quicktime-videossa käytetty videokoodekki on Sorenson Videon koodekki, jonka avulla on mahdollista pakata hyvälaatuista videota laajalla nopeusalueella. Parhaan tuloksen saavuttamiseksi Sorenson video hyödyntää vaihtuvanopeuksista koodausta. Vaihtuvanopeuksisessa koodauksessa video koodataan kahdessa vaiheessa. Ensimmäisessä vaiheessa koodekki käy videon läpi arvioiden, miten paljon bittejä kunkin kohdan koodaamiseen tarvitaan. Tämän jälkeen tavoitekapasiteetti jaetaan videossa siten, että paljon liikettä sisältävien kohtien koodaamiseen käytetään enemmän kapasiteettia kuin vähän liikettä sisältävien kohtien koodaamiseen. Tällöin lopullisessa bittivirran kaistanleveys vaihtelee koodekille asetettujen rajojen puitteissa ja videon laatu on säilyy tasaisena nopeissakin kohtauksissa. [40]

RealVideo

Real Networksin kehittämä RealVideo on erityisesti Internet-käyttöön soveltuva virtaava videoformaatti. RealVideon tukemat tiedonsiirtonopeudet ovat 20 - 220 kbps. RealVideossa sovellukset voivat vaihtaa tarvittaessa lähetysnopeutta kesken esityksen. Realin koodekit keskittyvät hyvin alhaisille bittinopeuksille, eivätkä suuremmilla

bittinopeuksilla pärjää kilpailijoilleen. Alle 100 kbps:n bittinopeuksilla koodatut videot jäävät laadultaan väistämättä melko heikoiksi, joten käytännössä ne soveltuvat vain trailereihin ja muihin lyhyisiin videon pätkiin. Näissä sovelluksissa se kuitenkin on erittäin suosittu media, koska RealVideo tarjoaa laajan tuen erilaisille alustoille. Real videon tukee virtaavaa dataa sekä progressiivista siirtoa. [40]

Windows Media

Windows Media Format on Microsoftin kehittämä multimedian tallennusmuoto. Windows Median videokodekki pohjautuu MPEG-4:ään, mutta sisältää laajennuksia, jotka parantavat kuvanlaatua. Kodekki soveltuu hyvin Internetin ja Windows Media Serverin kanssa käytettynä tukee virtaavaa videota, sekä HTTP:n päällä tapahtuvaa progressiivista videota. Windows Media Server tukee skaalautuvaa videota, jossa palvelimella on samasta videosta useita versioita, jotka on koodattu eri nopeuksilla. Esityksen alussa palvelin- ja asiakassovellus valitsevat parhaiten verkkoon mahtuvan version. Järjestelmä tukee kuitenkin vain yhtä audioraitaa, joten kaikki yhteydet jakavat samalla nopeudella koodatun äänen. Microsoft lupaa, että Windows Media mahdollistaa lähes DVD-tasaisen videolaadun 750 kbps nopeudella käyttäen 640 x 480 pikselin resoluutiota 24 kuvaa sekunnissa. VHS-tasaisen videon vaatimuksiksi luvataan 250 kbps. Nimensä mukaisesti Windows Media on ainoastaan käytettävissä Microsoft Windows:issa, mikä rajoittaa Windows Median käyttöä.

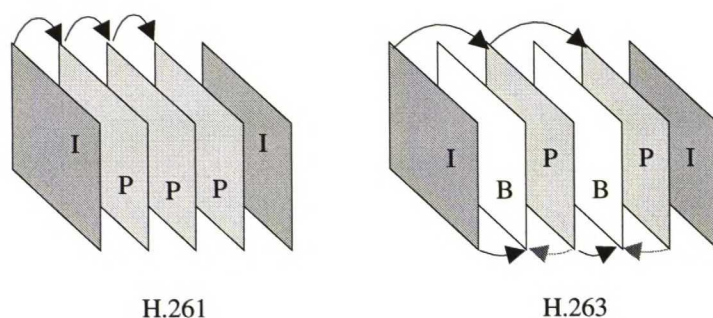
4.2.3 Videokonferenssi

Videokonferenssisovellusten oleellisin ero verrattuna muihin multimediasovelluksiin on reaaliaikaisuus. Yleisimmät videokonferenssistandardit ovat ITU-T:n määrittelemät H.261 [41] ja H.263 [42]. H.261-standardi määrittelee videokodekin videokonferenssiin ISDN-linjoilla, joiden kapasiteetti on $p \times 64$ Kbps, jossa p on välillä 1-30. Videon koodaus hyödyntää videossa oleva aikaredundanssia aggressiivisesti ja käyttää pitkiä ennustavasti koodattujen erotuskuvien sarjoja. Erotuskuvat sisältävät ainoastaan muutokset edelliseen kuvaan, ja pienet muutokset jätetään koodaamatta kokonaan.

H.261 oli alun perin suunniteltu häiriöttömään ISDN-verkkoon, joten sen soveltaminen Internetissä edellyttää pakettikatojen kompensointia. Turlettin ja Huiteman [43] esittelemässä menetelmässä H.261:stä ajetaan RTP:n päällä ja kadonneiden pakettien aiheuttamat virheet korjataan peruskuvien avulla. Vaihtoehtoisesti voidaan käyttää pelkkiä peruskuvia tai erityisellä protokollalla voidaan pyytää palvelinta päivittämään

virheellinen kohta uudella peruskuvalla. Internet ei myöskään rajoita yhteysnopeutta kiinteisiin rajoihin, joten siellä voidaan käyttää vaihtuvanopeuksista koodausta. Standardin mukainen maksiminopeus on 2 Mbps, tosin Internetissä maksiminopeutta on syytä rajoittaa alemmaksi.

Standardi soveltuu kuitenkin huonosti alle 64 kbps:n kanaville, joten modeemikäyttäjien tarpeisiin on kehitetty tehokkaampi H.263 -standardi. H.263 on laskennallisesti huomattavasti vaativampi kuin edeltäjänsä, koska se hyödyntää MPEG-koodauksen tapaan kaksisuuntaisesti ennustettuja B-kuvia. Erillisten B-kuvien sijaan H.263 käyttää PB-kehystä, jossa molemmat kuvat lähetetään samassa kehyksessä, mutta B-kuva esitetään ennen P-kuvaa. Langattomien kanavien tarpeisiin on olemassa H.263/M -standardi, joka toimii luotettavasti myös häiriöherkillä kanavilla. Standardien käyttämiä kuvia on havainnollistettu kuvassa 4.2. [1]



Kuva 4.2 Videokonfressin pakkautuminen.

Berliinin yliopistossa suoritetuissa mittauksissa testattiin H.263-standardin toimintaa koodaamalla täyspitkistä videoelokuvista digitoitua signaalia eri nopeuksilla. Taulukossa on esitetty tuloksia muutamien elokuvien osalta. Vaihtuvanopeuksisella koodauksella koodatuissa videoissa kooderi koodasi kaikki kuvat käyttäen I- ja PB-kehyskiä. Tehokkaammilla pakkaussuhteilla havaittiin, että pakkaustiheyden kasvaessa yhä useampi kuva jäi koodaamatta. [44]

Tavoitenopeus	Elokuva	Keskimääräinen kaistanleveys	Huippu/ keskiarvo
64 kbps	Jurassic Park	64 kbps	5,7
64 kbps	First Contact	64 kbps	5,8
Vaihtuva	Jurassic Park	450 kbps	7,7
Vaihtuva	First Contact	170 kbps	13,0

Taulukko 4.1. H.263 koodatun videon kaistanleveys.

4.2.4 Audio

Audiosignaalin siirto on videosignaaliin verrattuna huomattavasti helpompaa. Ihmisen kuuloaisti rajoittuu taajuusalueelle, joka ulottuu paristakymmenestä hertsistä kahteenkymmeneen kilohertsiin. Nyquistin näytteistysteoreeman mukaisesti 20 kHz:n signaalin toistamiseksi näytteistys on suoritettava vähintään 40 kHz:n tarkkuudella. Yleinen musiikin toistoon käytetty näytteistystaajuus, jota käytetään mm. CD-levyillä, on 44.1 kHz 16 bitin tarkkuudella. Tällöin pakkaamattoman stereosignaalin kaistanleveys on 1,4 Mbps. Normaalin puheäänien siirrossa voidaan käyttää edellistä pienempää tarkkuutta, ja digitaalisessa puhelinverkossa on päädytty käyttämään 8 kHz:n näytteistystaajuutta ja 8 bitin logaritmisesti koodattuja näytteitä. Tällä G.711-standardin mukaisella pulssikoodimodulaatiolla (PCM) päästään 64 kbps:n siirtonopeuteen ja 35 dB:n kohinasuhteeseen. Sen variaatiota on käytetty puheen siirtoon mm. Internetissä ja MBONE-verkossa.

Korkealaatuisen äänimaiseman luomiseksi tarvitaan yleensä vähintään kaksi äänikanavaa, ja elokuvissa yleistynyt monikanavaääni sisältää oikean ja vasemman kanavan lisäksi kaksi takakanavaa, etukanavan ja tehostekanavan matalia taajuuksia varten.

MPEG-audio koodataan 20 kHz taajuuskaistalla käyttämällä 44.1 kHz näytteistystaajuutta ja 16 bitin tarkkuutta, jolloin pakkaamattoman datan nopeus on 1.4 Mbps. Useimmat implementaatiot tarjoavat myös 32 ja 48 kHz:n näytteistystaajuudet. MPEG-audiopakkaus perustuu mm. psykoakustisen mallin hyödyntämiseen. Yksi psykoakustiikan sovellus pohjautuu ihmisen kuulossa havaittuun äänten peittymisilmiöön, ns. psykoakustiseen peittymiseen, joka ilmenee heikompien signaalien peittymisenä voimakkaan audiosignaalin viereisillä taajuusalueilla. Peittyneet signaalit voidaan siten pakata erittäin tehokkaasti vaikuttamatta lopputuloksen laatuun. Psykoakustinen pakkausalgoritmi analysoi signaalivoimakkuudet eri taajuusalueilla ja valitsee psykoakustisen mallin avulla kullekin taajuusalueelle optimaalisen bittimäärän ja kohinasuhteen, joiden

perusteella koodattu bittivirta muodostetaan. MPEG-1 tarjoaa kolme pakkaustasoa, joista kaksi ensimmäistä ovat prosessorikuormitukseltaan kevyempiä ja soveltuvat myös tosiaikaiseen koodaukseen. Kolmostason pakkaus on edellisiä raskaampi ja hyödyntää mm. äänidatan muuntamista taajuusalueelle. Kolmostason pakkauksella voidaan CD-levyn PCM-koodattu 1536 kbps stereoäänivirta koodattua lähes alkuperäisen veroisena 144 kbps nopeudella. MPEG-audiovirta sekoitetaan videodataan aikaleimojen avulla siten, että dekodderi voi erottaa audion ja videon toisistaan ennen dekoddausta ja esitystä. [1]

MPEG-2 -audio sisältää MPEG-1 -audion ja laajentaa sitä monikanavajärjestelmiä varten. MPEG-1:ssä on vain kaksi äänikanavaa, kun taas MPEG-2 voi sisältää jopa viisi äänikanavaa, joista kaksi ensimmäistä ovat yhteensopivia MPEG-1:n kanssa. MPEG-2 sallii myös alempien 24, 22.05 ja 16 kilohertsin näytteistystaajuuksien käytön. [1]

Internetissä suosituksi tullut musiikinlevitysmuoto, MP3-tiedostomuoto, käyttää MPEG:n kolmostason pakkausalgoritmia. Tyypillisesti MP3-tiedostot on Internetissä koodattu 128 kbps tai 196 kbps nopeudella, joilla saavutetaan musiikin kuuntelun kannalta riittävä äänenlaatu. Ääninäytteiden koodaamisessa käytetään usein myös alempia nopeuksia. MP3-muotoisen musiikin levitys ei varsinaisesti perustu virtaavaan dataan. MP3-yhteisöt ovat luoneet menetelmiä, joilla MP3-tiedostoja voidaan soittaa Internetissä progressiivisesti virtaavan datan tavoin. Tyypillisesti nämä menetelmät siirtävät MP3-tiedostoa TCP:n tai HTTP:n päällä ja puskuroivat dataa tiedonsiirtovirheiden kompensoimiseksi. TCP:n vuonhallintamekanismien avulla soittimet voivat hallita datan siirtoa ja musiikin toistoa.

Varsinaisista siirtomenetelmistä, joilla MP3:sia siirretään ei ollut juuri tietoa tarjolla. Menetelmien selvittämiseksi suoritettiin koe, jossa MP3-tiedostoja soitettiin MP3.com:in palvelimelta XMMS-soittimella [45]. Kaappaamalla palvelimen ja soittimen välistä liikennettä pystyttiin hahmottamaan, miten progressiivinen tiedonsiirto on soittimessa toteutettu, ja minkälaisia vaikutuksia mahdollisilla tiedonsiirtovirheillä tulisi olemaan. Yhteyden alussa soitin latsasi MP3-tiedostoa maksiminopeudella noin 10 sekunnin ajan täyttäen 32 kilotavun soittopuskurin, jonka jälkeen tahti hidastui 128 kilobitin nopeuden vaatimalle tasolle. Tiedonsiirto kulki HTTP:n päällä ja palvelin lähetti datan 1500 tavun paketeissa. Alussa paketteja saapui noin 20 kpl/s ja puskurin täytyttyä nopeus laski suurinpiirtein 10 pakettiin sekunnissa. Pakettien saapumisajat eivät olleet tasaisia, vaan ne tuntuivat tulevan pareittain. Soitin ohjasi pakettien saapumista kuittaamalla paketteja

sitä mukaa kun uutta dataa tarvittiin ja sääti tarpeen mukaan TCP:n lähetyssikkunan kokoa. Siirtovirheistä toipuminen tapahtui TCP:n uudelleenlähetyksen avulla. Pitemmät katkokset tiedonsiirtoyhteydessä tai kaistanleveyden lasku liian pieneksi aiheuttivat soittoon katkoksen, jonka jälkeen soitin joutui uudelleen täyttämään soittopuskurin. Katkoksen jälkeen soitto useimmiten jatkui samasta kohtaa, mihin se oli jäänyt. Pitemmät katkokset saattoivat katkaista kappaleen soiton.

4.2.5 Vuorovaikutus ja synkronointi

Vuorovaikutuksen kannalta on oleellista, että multimediaesityksen osat latautuvat synkronoidusti. Tämä tarkoittaa sitä, että erilaiset mediaelementit on siirrettävä hallitusti yhtä aikaa. Esimerkiksi Quicktime ja MPEG-4 multipleksaavat eri elementit yhteen virtaan, jolloin elementtien siirron synkronointi hoidetaan koodausvaiheessa. Tällöin multimediaesitys muistuttaakin enemmän interaktiivista videota kuin erillistä mediakomponenteista koostettua multimediaesitystä.

HTML:ään ja SMIL:iin pohjautuvissa esityksissä eri mediaelementit haetaan erikseen. SMIL tarjoaa mahdollisuuden synkronoida eri elementtien siirto, mutta HTML ei tarjoa vastaavaa mahdollisuutta. Pelkkä synkronointi ei aina riitä, sillä elementit on myös siirrettävä verkon yli samanaikaisesti. Tämä saattaa aiheuttaa ongelmia erityisesti pienikapasiteettisilla yhteyksillä, joiden kapasiteetti ei riitä siirtämään kaikkea sovelluksen haluamaa dataa. Esimerkiksi sovelluksen esittämä video voi syödä valtaosan tiedonsiirto-kapasiteetista, jolloin kuvien lataaminen hidastuu huomattavasti. Vastaavasti muun datan siirto voi aiheuttaa virheitä audion tai videon siirtoon.

4.2.6 XML-pohjaiset multimediapalvelut

XML-pohjaiset palvelut tarjoavat HTML:ään verrattuna huomattavasti laajemmat mahdollisuudet sovittaa palvelujen multimediakomponentit erilaisille päätelaitteille ja verkkoympäristöille kuin HTML-pohjaiset palvelut. XML-sivu voi tarjota datasta useita erilaisia versioita, joista XML-selain voi valita soveltuvan version. Multimedian kannalta tämä tarkoittaa sitä, että samasta multimediapalvelusta voi olla tarjolla useita erikapasiteettisia versioita. Selaimen ja palvelimen yhteistoiminnan avulla voidaan hyödyntää verkon tarjoamia palveluita mahdollisimman tehokkaasti. Esimerkiksi langattomassa verkossa voidaan normaalisti käyttää esityksestä kapeampikaistaista virheenkorjausta tukevaa versiota, mutta palvelunlaatua tukevassa verkossa käytettävä versio voidaan valita tarjolla olevan kapasiteetin mukaan.

5 Multimedian tehokas siirto langattomassa verkossa

Multimedian tehokas siirto langattomassa verkossa edellyttää, että verkko pystyy täyttämään multimedian palvelutarpeet. Virheellisesti siirretty multimedia hukkaa kaistanleveyttä, koska kulutetun kaistan määrästä huolimatta käyttäjien palvelutarpeita ei pystytä täyttämään.

5.1 Multimedian palvelutarpeet

Pääosa multimediasovelluksista siirtää multimediaa yhteen suuntaan paluuliikenteen sisältäessä ohjausdataa. Tällaisia sovelluksia ovat mm. Internet radiot, etävideonsovellukset ja multimediakomponentteja sisältävät WWW-sivut. Käyttäjän kannalta viive ei ole ratkaiseva tekijä, joten sovellukset pystyvät mukautumaan viiveen vaihteluun puskuroimalla tulevaa dataa. Keskeiset vaatimukset kohdistuvat tällöin kaistanleveyteen ja pakettikatoon.

Kaksisuuntaista reaaliaikaista dataa hyödyntävät sovellukset, kuten videokonferenssi ja IP-puhelu, vaativat ennen kaikkea pientä pakettikatoa, viivettä ja viiveen vaihtelua. Luonteva keskustelu edellyttää, että viive pysyy alle 100 millisekunnissa. Lisäksi puheen ymmärtäminen edellyttää, että puheen siirrossa ei tule liikaa virheitä. Videokonferensseissa käyttäjät ovatkin usein valmiita tinkimään videon laadusta, jotta puhe saadaan siirrettyä tarpeeksi laadukkaasti.

5.1.1 Kaistanleveys

Se, miten paljon mediaelementtejä on pakattava siirtoa varten, riippuu kaistanleveydestä. Tyypillisesti videon, audion ja kuvien siirrossa käytetään tehokkaita häviöllisiä pakkausmenetelmiä, joiden avulla niiden tarvitsema kaistanleveys pienenee huomattavasti.

Multimediatatan siirrossa käytetyn virtaavan datan toiminta perustuu siihen oletukseen, että verkon kaistanleveys riittää siirtämään tietyn määrän dataa jatkuvasti. Tämä kaistanleveyden jatkuvuus on sovelluksille tärkeää. Sovelluksen odottaman viiveen vaihtelun puitteissa tapahtuva kaistanleveyden lyhytaikainen lasku on hyväksyttävää. Pidemmistä kaistanleveyden laskuista aiheutuu välttämättä virheitä, koska sovelluksilla ei ole normaalisti mitään mahdollisuutta varautua kapasiteetin laskuun. Toisaalta sovellukset pystyvät mukautumaan erilaisiin olosuhteisiin käyttämällä tehokkaampia pakkausmenetelmiä, mikäli tieto esimerkiksi laskevasta kapasiteetista on etukäteen tarjolla.

5.1.2 Pakettikato

Pakettikato on multimediasovellusten kannalta erittäin ongelmallista, koska virtaavaa dataa ei tyypillisesti ehditä lähettämään uudelleen ja jokainen kadonnut paketti aiheuttaa virheen esitykseen. Useat pakkausalgoritmit käyttävät hyväksi aiemmin lähetettyä dataa, jolloin yksittäinen virhe voi vaikuttaa esitykseen pitkänkin ajan. Kumulatiiviset virheet voivat nopeasti alentaa esityksen laatua huomattavasti.

Uudemmissa standardeissa, kuten MPEG-4:ssä, käytetään mm. ennakkoivaa virheenkorjausta, virheentunnistusta ja virheenpiiloitusta minimoimaan siirtovirheiden vaikutusta esityksen laatuun. Nämä tekniikat kuitenkin keskittyvät yksittäisten bittivirheiden korjaamiseen. Kokonaisten pakettien katoaminen on huomattavasti vaikeampi korjata, joten pakettikadolla on huomattava vaikutus käyttäjän kokemaan latuun.

5.1.3 Viive

Yksisuuntaisessa multimediatatassa viiveellä ei ole suoranaista vaikutusta esityksen laatuun, mutta interaktiivisia komponentteja sisältävässä esityksessä interaktiivisuus alkaa kärsiä, jos viive kasvaa liian suureksi. Erityisesti interaktiota sisältävät sovellukset, kuten videokonferenssi ja IP-puhelu, ovat herkkiä viiveen kasvulle. Jotta osallistujien välinen vuorovaikutus säilyisi luonnollisena, on kokonaisviiveen oltava alle 100 ms [46]. Sovellusten käyttämät koodekit ja puskurointi aiheuttavat oman viiveensä siirtoon, joten verkon aiheuttaman viiveen tulisi jäädä alle 50 ms:iin.

5.1.4 Viiveenvaihtelu

IP-verkossa paketit kulkevat yksittäin ja niiden siirtoviiveet vaihtelevat. Vaihtelu johtuu mm. erilaisista reitityspäätöksistä ja muun liikenteen määrästä. Kevyesti kuormitetussa verkossa siirtoviiveet ovat suhteellisen pieniä, kun taas raskaasti kuormitetussa verkossa siirtoviiveet nousevat.

Sovellukset kuitenkin odottavat saavansa dataa tasaisena virtana, joten niiden on tasoitettava siirtoviiveissä tapahtuva vaihtelu puskuroimalla saapuvaa dataa. Periaatteessa sovellus voi varautua jopa useiden sekuntien mittaisiin viiveenvaihteluihin, mutta joutuu samalla hyväksymään suuremman viiveen. Puskuroinnista aiheutuva viive lisää kokonaisviivettä viiveenvaihtelun verran, joten sovellusten toiminta edellyttää rajoitettua viiveenvaihtelua, jonka suuruus on sovellusten tiedossa.

5.2 Palvelutarpeiden tyydyttäminen

Multimedian vaatimien palvelutarpeiden tyydyttämisessä ehdottomasti tärkein tekijä oli kaistanleveyden takaaminen kullekin multimediasynteille. Omalla varatulla kaistallaan liikennöivä sovellus ei kärsi satunnaisista ylikuormituksesta johtuvista pakettikadoista. Samalla viiveen ja viiveenvaihtelu pysyvät hallinnassa.

Kappaleessa 3 esiteltiin erilaisia tekniikoita palvelunlaadun toteuttamiseksi. Erityisesti CBQ-jonotukseen perustuva vaihtoehto vaikutti lupaavalta toteutusvaihtoehdolta. Yulian Wang esitteli diplomityössään [47] CBQ-algoritmiin perustuvan ratkaisun. CBQ-algoritmin avulla liikenne jaettiin kahteen pääluokkaan, joista toista käytettiin varatulle liikenteelle ja toista muulle liikenteelle. Varatulla kaistalla liikennöivät sovellukset joutuivat varaamaan käyttämänsä kaistanleveyden, ja siten varatun kaistan käyttöä voitiin valvoa ja välttyä ylikuormitustilanteilta. Molemmat pääluokat voitiin edelleen jakaa aliluokkiin erilaisia medioita varten.

Verkon kapasiteetti jaettiin CBQ-luokkien kesken siten, että valtaosa kapasiteetista annettiin varattujen luokkien käyttöön. Vapaan liikenteen luokille jätettiin vain hyvin kapeat kaistat. Näin varatulle liikenteelle voitiin antaa etusija verkon käytössä, mutta samalla vältyttiin vapaalla kaistalla liikennöivien yhteyksien täydelliseltä nälkiintymiseltä ylikuormitustilanteissa. Käyttöasteen maksimoimiseksi luokille annettiin mahdollisuus lainata muilta luokilta käyttämättä jäänyttä kapasiteettia. Lainaaminen tapahtui luokille asetettujen prioriteettien perusteella siten, että ylemmän prioriteetin liikenteellä oli ensisijainen oikeus lainata yli jäänyttä kapasiteettia.

Teoriassa CBQ-algoritmiin pohjautuva palvelunlaatu vaikutti toimivalta ratkaisulta, mutta sen toimivuus käytännössä oli epävarmaa. Toimivuuden selvittämiseksi CBQ-algoritmin toimintaa verrattiin FIFO-algoritmin toimintaan simulaatioiden avulla.

6 Simuloinnit langattomassa verkossa

6.1 Simulaatioiden tavoitteet

Simulaatioiden tavoitteena oli selvittää CBQ-pohjaisen palvelunlaadun vaikutusta langattomassa verkossa siirrettävään multimediadataan. Samalla pyrittiin tarkistamaan, miten Yulian Wangin [47] ehdottama CBQ-algoritmiin perustuva malli skaalautuu aikaisempia testejä suurempiin langattomiin verkkoihin sekä siirrettävän datan määrän kasvuun. Skaalautuvuutta ja algoritmin toimintaa arvioitiin vertaamalla CBQ-jonotuksella ja FIFO-jonotuksella saatuja tuloksia keskenään erilaisissa kuormitustilanteissa.

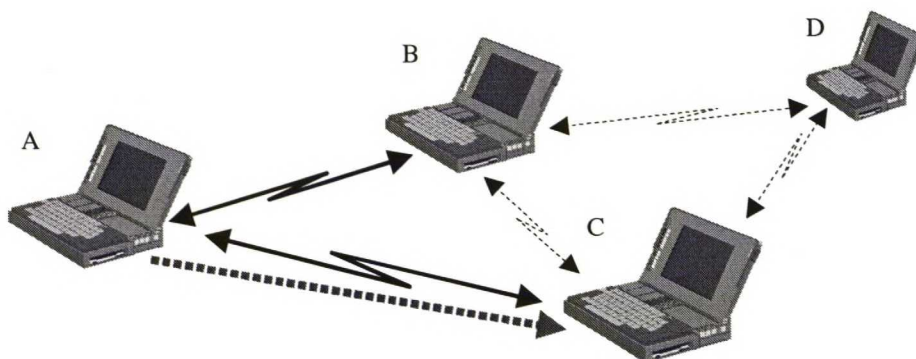
6.2 Testausympäristö

Langattoman verkon simulointiin käytettiin Virtual InterNetwork Testbed -projektin [48] kehittämää Ns-verkkosimulaattoria. Ns-simulaattori on diskreettitapahtumasimulaattori, joka on suunniteltu tietoverkkojen ja protokollien tutkimukseen. Ns-simulaattori valittiin simulointivälineeksi, koska se on laajalti käytössä tutkimusyhteisöissä ja tukee langattomien verkkojen simulointia. Lisäksi simulaattoriin on saatavilla valmiit toteutukset tunnetuimmista tiedonsiirtoprotokollista ja jonotusalgoritmeista.

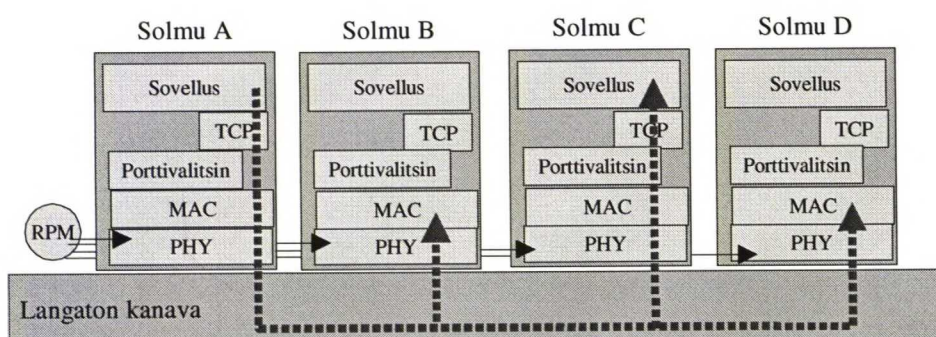
6.2.1 Langattoman verkon mallinnus

Simulaattorin tuki langattomille verkoille perustuu CMU Monarch -projektin [49] kehittämille laajennuksille. Laajennukset tarjoavat yksityiskohtaisen mallin langattomasta mediasta käyttäen mm. radioaaltojen etenemismalleja, antennoja, linkki- ja MAC-kerroksen protokollia sekä ad hoc -reititysprotokollia.

Simulaattorin malli langattomasta verkosta muodostuu *solmuista*, jotka sisältävät mm. tiedon sijainnistaan ja jotka voivat liikkua paikasta toiseen. Solmut liittyvät toisiinsa *kanavien* välityksellä. Kanavalle lähetetyt paketit välittyvät kaikille samalla kanavalle oleville solmuille. Samassa langattomassa verkossa olevat päätelaitteet sijaitsevat samalla kanavalla. Kuvissa 6.1 ja 6.2 on esitetty simulaattorin tapa mallintaa langatonta verkkoa neljän koneen muodostaman verkon avulla. Kuvassa 6.1 kone A yrittää lähettää dataa B:lle. Kantoalueella olevat B ja C kuulevat lähetyksen, jolloin B jatkaa viestin vastaanottamista ja lähettää sen edelleen kohdesovellukselle. C hylkää saapuvan datan, koska data ei ollut tarkoitettu sille. Kantoalueen ulkopuolella oleva kone D ei kuule lähetystä lainkaan, joten lähetyksellä ei ole vaikutusta sen toimintaan.



Kuva 6.1. Neljän tietokoneen muodostama langaton verkko, jossa A lähettää dataa B:lle.



Kuva 6.2. Simulaatiomalli samasta verkosta.

Kuvassa 6.2 sama on esitetty simulaattorissa toteutetun mallin mukaisesti. Solmuissa paketit kulkevat protokollapinon läpi, jossa vastaanotetut paketit menevät ensin fyysiseen kerrokseen (PHY). Fyysinen kerros määrittää radioaaltojen etenemismallin (RPM) avulla vastaanotetun paketin signaalinvoimakkuuden ja vertaa sitä sitten *havaitsemiskynnykseen* määrittääkseen, pystyykö solmu havaitsemaan kanavalle lähetettyä pakettia. Havaittujen pakettien signaalinvoimakkuutta verrataan edelleen *vastaanottokynnykseen*, joka määrittelee onnistuneen vastaanoton vaatiman signaalinvoimakkuuden. Jos vastaanotetun signaalin voimakkuus alittaa kynnyksen, paketti merkitään virheellisesti vastaanotetuksi. Kaikki paketit toimitetaan kuitenkin MAC-kerrokselle. MAC-kerros hylkää virheellisesti vastaanotetut paketit sekä paketit, joita solmu ei pystynyt vastaanottamaan.

Simulaattorin toteutus IEEE 802.11 -standardin mukaisesta MAC-kerroksesta käyttää RTS/CTS/Data/ACK -mallia pakettien välittämiseen ja sisältää uudelleenlähetykset virheellisesti toimitetuille paketeille. Broadcast-paketit lähetetään ilman uudelleenlähetystä.

6.2.2 Radioaaltojen etenemismallit

Ns-simulaattori tarjoaa langattomien verkkojen mallintamiseen kolme etenemismallia. Pallosäteily-etenemismalli (*free space model*) ennustaa radioaaltojen etenemistä vapaassa tilassa. Maaheijastusmalli (*two-ray ground*) -etenemismalli soveltuu tasaiseen avoimeen maastoon. Katvemalli (*shadowing model*) mallintaa etäisyyden lisäksi myös satunnaisten esteiden vaikutusta radioaaltoihin. Koska simuloinneissa on tarkoitus huomioida hetkellisten katveiden vaikutus yhteyteen, soveltuu katvemalli parhaiten simulointeihin. Seuraavassa on esitelty lyhyesti simuloinneissa käytetyt etenemismallit. Käsittely pohjautuu Rappaportin [50] kirjaan.

Pallosäteily -etenemismalli

Pallosäteily -etenemismalli olettaa, että lähettimen ja vastaanottimen välillä on ainoastaan yksi, suora, esteetön reitti. Tällaisessa ideaalisessa tapauksessa vastaanotetun signaalin voimakkuus etäisyydellä d voidaan laskea seuraavasti:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4 \pi d)^2 L}, \quad (6.1)$$

jossa P_t on lähetyn signaalin teho. G_t ja G_r ovat lähettimen ja vastaanottimen antennien vahvistukset. L on järjestelmähäviö ja λ on aallonpituus. Simuloinneissa ns-simulaattorilla käytetään yleensä arvoja $G_t=G_r=L=1$.

Malli kuvaa lähettimen kantoalueen ympyräksi, jonka sisällä sijaitsevat vastaanottimet kuulevat kaikki lähetetyt paketit. Ulkopuolella olevat vastaanottimet puolestaan eivät kuule ainoatakaan pakettia.

Katvemalli

Todellisuudessa vastaanotetun signaalin voimakkuus tietyllä etäisyydellä lähettimestä on satunnainen. Tämä johtuu mm. signaalin heijastumisesta ympäristössä ja satunnaisten esteiden aiheuttamista vaimennuksista.

Katvemalli (*shadowing model*) koostuu kahdesta osasta. Ensimmäinen osa, jota kutsutaan tiehäviömalliksi, määrittää keskimääräisen vastaanottotehon $P_r^k(d)$ etäisyydellä d . Teho lasketaan suhteessa vertailuetäisyydellä d_0 mitattuun tehoon $P_r(d_0)$ seuraavasti:

$$\frac{P_r(d_0)}{P_r^k(d)} = \left(\frac{d}{d_0} \right)^\beta \quad (6.2)$$

Kaavassa β on tiehäviövakio, joka yleensä määritetään kenttämittauksilla. Taulukossa 6.1 on mainittu muutamia tyypillisiä vakion β arvoja. Vertailuteho $P_r(d_0)$ voidaan laskea esimerkiksi pallosäteilymallilla.

Tavallisesti tiehäviö ilmoitetaan desibeleinä, jolloin kaavasta (6.2) saadaan

$$\left[\frac{P_r^k(d)}{P_r(d_0)} \right]_{dB} = -10 \beta \log \left(\frac{d}{d_0} \right) \quad (6.3)$$

Katvemallin toinen osa mallintaa vastaanotetun signaalin voimakkuuden vaihtelua tietyllä etäisyydellä. Tähän käytetään satunnaismuuttujaa, jonka jakauma vastaa desibeleissä mitattuna Gaussin jakaumaa. Näin katvemallin mukaiseksi vastaanotetun signaalin tehoksi saadaan

$$\left[\frac{P_r^k(d)}{P_r(d_0)} \right]_{dB} = -10 \beta \log \left(\frac{d}{d_0} \right) + X_{dB} \quad (6.4)$$

jossa X_{dB} on Gauss-jakautunut satunnaismuuttuja, jonka odotusarvo on nolla ja keskihajonta σ_{dB} . Keskihajontaa kutsutaan myös katvehajonnaksi, ja sen arvo voidaan määrittää kokeellisesti. Taulukko 6.2 esittelee tyypillisiä σ_{dB} :n arvoja.

Ympäristö		β
Ulkoilma	Vapaa tila	2
	Katveinen kaupunkiympäristö	2.7 - 5
Sisätila	Suora näkyvyys	1.6 - 1.8
	Estynyt näkyvyys	4 - 6

Taulukko 6.1. Tiehäviövakion β tyypillisiä arvoja.

Ympäristö	σ_{dB} (dB)
Ulkoilma	4 - 12
Toimisto, kiinteäseinäinen	7
Toimisto, avorakenteinen	9.6
Tehdas, suora näkyvyys	3 - 6
Tehdas, estynyt näkyvyys	6.8

Taulukko 6.2. Katvehajontavakion σ_{dB} tyypillisiä arvoja.

6.2.3 Simulointiympäristö

Simulointiympäristö asennettiin ns2-simulaattorin paketista, jonka versionumero oli 2.1b8. Langattoman verkon käyttäytymistä simuloitiin katvemallin avulla. Katvemallin parametrit asetettiin vastaamaan langattoman verkon käyttöä kiinteäseinäisessä toimistossa, jossa näkyvyys tukiasemaan voi olla estynyt.

Simulaattorin langattoman verkon toteutus oli alunperin suunniteltu ad-hoc -verkkojen tutkimukseen, joten CBQ-algoritmin sovittaminen langattomaan verkkoon vaati pienten muutosten tekoa simulaattoriin. Muutokset kuitenkin rajoittuivat CBQ-jonotuksen vaatimien hallintarakenteiden lisäämiseen, eivätkä siten aiheuttaneet muutoksia simulaattorin rakenteeseen tai tulosten oikeellisuuteen. Simulaatioissa ei käytetty minkäänlaista varausprotokollaa, jolla CBQ-luokkien liikennettä olisi dynaamisesti hallittu. Yhteydet sidottiin tiettyihin CBQ-luokkiin virtatunnisteiden avulla eikä tukiasema mitenkään valvonut varausten toteutumista. Näin pystyttiin tutkimaan myös varatun luokan ylikuormittumisesta aiheutuvia tilanteita.

6.2.4 Simulaattorin validointi

Simulaattorin sisältämä validointiohjelma mahdollisti simulaattorin ohjelmallisen toimivuuden tarkistamisen simulointialustalla ja validoinnin ajo osoitti, että simulaattori toimi odotetulla tavalla. Validointiohjelman testit rajoittuivat kuitenkin vain mallien sisäisen toiminnan tarkistamiseen, eivätkä ottaneet mitenkään kantaa siihen, miten hyvin mallit soveltuvat yksittäisten reaalimaailman olosuhteiden kuvaamiseen. Koska tulosten soveltamisen kannalta on tärkeää pystyä edes jollakin tasolla arvioimaan sitä, miten hyvin simulaattorissa käytetyt mallit kuvaavat todellisen langattoman verkon olosuhteita, toistettiin joukko langattomassa verkossa suoritettuja testejä simulaattorilla.

Validointimateriaaliksi oli tarjolla Yulian Wangin [47] langattomassa verkossa suorittama testisarja, joka tutki CBQ-algoritmin soveltamista langattomaan verkkoon Mobile-IP:n alla. Testit rajoittuivat yhden tukiaseman alueella tapahtuvaan liikenteeseen ja

keskittyivät tasaisten bittivirtojen ja TCP-virtojen käyttäytymiseen erilaisissa ylikuormitustilanteissa. Simulaattorin tuki Mobile-IP:lle sekä langattoman verkon solmuja sisältäville sekaverkoille vaikutti kuitenkin sen verran epämääräiseltä, että simulointimallissa päädyttiin simuloimaan pelkästään ilmarajapinnan liikennettä. Tätä ratkaisua tuki myös se, että vertailtavissa testeissä Mobile-IP:n tarjoamaa liikkuvuutta ei hyödynnetty. Mobile-IP oli käytössä ainoastaan pakettien reitityksessä langattoman ja langattoman verkon välillä. Lisäksi testeissä käytetyt kaistanleveydet olivat huomattavasti pienempiä kuin langattoman ja Mobile-IP:n kapasiteetti. Mobile-IP:n käyttämä datan tunnelointi aiheuttaa pienen kapasiteetin menetyksen Mobile-IP:n yli siirrettyyn dataan, mutta testeissä käytetyillä paketeilla tunneloinnin vaikutus jäi häviävän pieneksi.

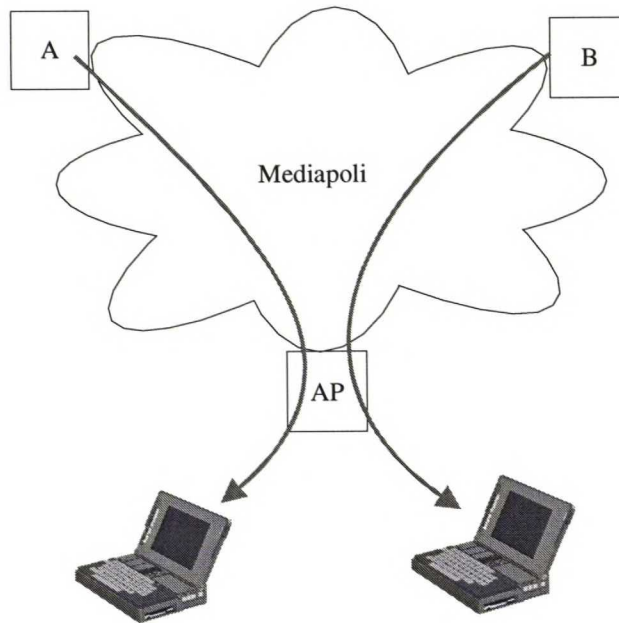
Alkuperäisissä testeissä verkon toimintaa mitattiin Iperf-mittausohjelman [51] avulla, joka mittasi TCP- ja UDP-liikenteen läpimenoa sekä pystyi mittaamaan saavutetun kaistanleveyden, huojunnan ja pakettikadon. Simulaattorissa mittausohjelman vastineena käytettiin TCP- ja UDP- liikennettä kehittäviä liikennegeneraattoreita, jotka lähettivät liikennettä langattoman verkon tukiasemasta. Alkuperäisessä testiympäristössä paketit kulkivat ennen siirtymistään langattomaan verkkoon lähiverkon läpi, joten liikennegeneraattorit asetettiin luomaan pientä satunnaisuutta pakettien lähetyaikoihin. Ilman satunnaisuutta pakettien synkronoituminen simulaattorissa olisi vääristänyt tuloksia. UDP- ja TCP-virtojen kaistanleveys ja pakettikato mitattiin periodisesti ja talletettiin tiedostoihin simulaation aikana. Viiveiden jakautuminen UDP-paketeilla voitiin selvittää simulaattorin tuottamista lokitiedostoista laskemalla pakettien saapumis- ja lähtöaikojen välinen erotus.

Simulointien tuloksia verrattiin Yulian Wangin saamiin tuloksiin [47]. Testeissä, joissa verkon ylikuormitus oli pientä, tuloksissa ei ollut merkittäviä eroja. Ongelmia sen sijaan ilmeni testeissä, joissa verkkoa ylikuormitettiin raskaasti. Yulian Wangin tekemät testit osoittivat voimakkaan kokonaiskapasiteetin laskun, kun samassa verkossa oli useita yhtäaikaista 1,5 Mbps:n yhteyksiä. Sama ilmiö esiintyi jo yhdellä yksittäisellä 2 Mbps yhteydellä ja useampi 2 Mbps:n yhteys aiheutti kokonaiskapasiteetin kaoottisen vaihtelun. Simulaattorissa vastaavia ilmiöitä ei havaittu. Simulaatioissa langaton verkko pystyi toimimaan huomattavasti tehokkaammin ylikuormitustilanteissa, eikä kokonaisläpimenoa voitu havaita vastaavaa laskua. Malli rajoitti läpimenon langattomalle verkolle määritettyyn maksimikapasiteettiin, mutta ei aiheuttanut ylimääräistä läpimenon laskua. TCP-yhteyksillä läpimenon lasku on normaalia, mutta UDP-yhteyksillä se vaikutti epäi-

lyttävältä. Etenkin kun jo yksi 2 Mbps yhteys riitti tukkimaan verkon ja aiheuttamaan läpimenon laskun, vaikka 2 Mbps nopeudella toimivan radioverkon pitäisi kyetä tarjoamaan 1,6 Mbps läpimeno UDP-datalle.

6.2.5 Läpimenon laskun tutkiminen

Läpimenon laskua tutkittiin edelleen HUT Mediapolin verkossa, joka tarjosi alkuperäistä testiympäristöä vastaavat olosuhteet. Testikoneina käytettiin kahta langattomaan verkkoon liitettyä kannettavaa Linux-tietokonetta, jotka oli varustettu Wavelanin 2 Mbps:n WLAN-korteilla. Koska haluttiin keskittyä nimenomaan langattoman verkon käyttäytymiseen, koneissa ei käytetty Mobile-IP:ta, vaan WLAN-kortit liikennöivät ad hoc -tilassa keskenään ilman tukiasemaa. Tavoitteena oli samalla saada aikaisempia testejä parempi käsitys siitä, miten viiveet ja pakettikadot jakautuvat langattomassa verkossa. Aikaisemmissa testeissä käytetty Iperf-ohjelmisto ei täyttänyt näitä kriteereitä, joten testejä varten kehitettiin oma testiohjelma. Käytetty verkko on havainnollistettu kuvassa 6.3.



Kuva 6.3. Läpimenon laskun tutkimisessa käytetty verkko.

Testiohjelma koostui lähettimestä, joka lähetti UDP-pakettivirtaa, sekä vastaanottimesta, joka mittasi ja talletti pakettien viiveet ja katoamiset sekä laski keskimääräisen kaistanleveyden. Lähetetyt UDP-paketit varustettiin järjestysnumeroilla ja aikaleimoilla,

joiden avulla vastaanotin pystyi laskemaan pakettikohtaiset viiveet. Käyttöjärjestelmän vaikutus pakettien viiveisiin pyrittiin minimoimaan käyttämällä mikrosekuntien tarkkuudella otettuja aikaleimoja viiveiden laskemiseen. Testikoneiden kellojen todettiin käyvän hieman eri nopeuksilla. Lähettäjän ja vastaanottajan kellojen epätarkkuuden kompensoimiseksi ohjelmiin lisättiin algoritmi, joka mittasi lähettäjän ja vastaanottajan kellojen välisen nopeuseron ja korjasi vastaanotettujen pakettien lähetysaikoja vastaavasti.

Testisarja koostui viidestä testistä, joilla yritettiin siirtää UDP-dataa 1000, 1500, 2000, 2500 ja 3000 kbps:n nopeuksilla. Testeissä käytettiin 4 millisekunnin lähetysviivettä ja vaihdeltiin pakettien kokoa. Testeissä paketteja lähetettiin, kunnes 5000 pakettia oli vastaanotettu. Testien keskeiset tulokset on esitetty taulukossa 6.3. Lähetysnopeus on testiohjelman tuottaman liikenteen nopeus ja saavutettu läpimeno on vastaanotetun datan nopeus. Saapumisviive ilmoittaa peräkkäisten pakettien keskimääräisen ja maksimaalisen vastaanottoviiveen. Tuloksista voidaan havaita läpimenon hienoinen kasvu pakettikoon suurentuessa. WLAN-verkossa, jonka siirtonopeus on 2 Mbps, MAC-otsikko lähetetään aina 1 Mpbs nopeudella ja suurempaa siirtonopeutta käytetään datan siirtoon. Tällöin pakettikoon suurentuminen parantaa radioliikenteen tehokkuutta.

Hyvässä, häiriöttömässä radioverkossa voidaan tulosten mukaan saavuttaa noin 1,6 Mbps:n läpimeno. Yulian Wangin testeissä ilmennyttä kokonaiskapasiteetin laskua ylikuormitustilanteessa ei havaittu, joten todennäköisesti kapasiteetin laskuun vaikuttivat langattoman verkon ulkopuoliset tekijät.

Lähetysnopeus kb/s	Pakettikoko tavuina	Saavutettu läpimeno kb/s	Saapumisviive ms		Viiveen varianssi
			kesk	maks	
1000	500	1000	4,1	9	0,063
1500	750	1450	4,1	9,0	0,053
2000	1000	1565	5,1	10,0	0,087
2500	1250	1585	6,3	19	0,760
3000	1450	1620	7,2	28	1,110

Taulukko 6.3. Läpimeno langattoman linkin yli 4 ms:n lähetysviiveellä.

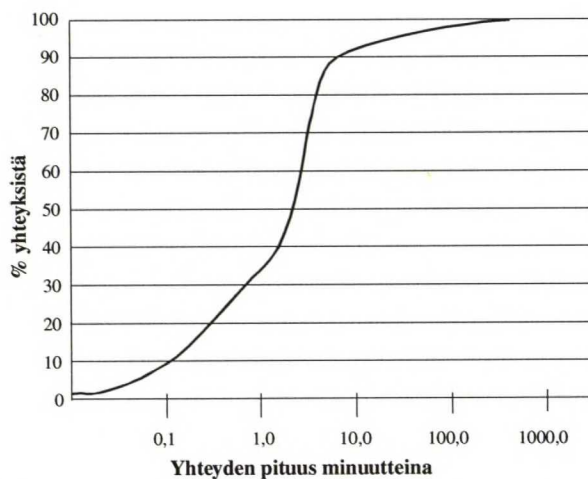
6.3 Multimedialiikenteen mallintaminen

Multimedialiikenteen mallintamisessa lähdettiin liikkeelle Internetissä tehdyistä mittaustuloksista ja pyrittiin niiden avulla hahmottamaan, miten käyttäjät hyödyntävät

multimediapalveluita. Tulosten pohjalta kehitettiin edelleen käyttäytymismalleja, joilla ennustettiin käyttäjien aiheuttamaa multimedialiikennettä langattomassa verkossa.

6.3.1 Multimedialiikenteen luonne

Washington yliopiston verkossa suoritettut mittaukset [52] ovat osoittaneet, että valtaosa multimediyhteyksiä siirtää virtaavaa dataa alhaisella siirtonopeudella (alle 56 kbps), on kestoaltaan lyhytaikaista (alle 5 minuuttia). Samalla ilmeni, että useimmat käyttäjät siirsivät yhteyden aikana suhteellisen vähän dataa, alle 1 megatavua. Toisaalta muutamat aktiivikäyttäjät kuluttivat huomattavasti enemmän resursseja pitkällä ja laajakaistaisilla yhteyksillä. Huomattavaa oli, että myös nopeilla verkkoyhteyksillä varustetut käyttäjät tyytyivät alhaisiin nopeuksiin multimediyhteyksissä. Tämä johtunee siitä, että modeeminopeuksille soveltuvaa multimediaa on verkossa tarjolla enemmän kuin laajakaistaista multimediaa. Matala siirtonopeus viittaa vahvasti myös siihen, että audio on videota suosittu media. Esimerkiksi internetradiot toimivat useimmiten 20 - 56 kbps:n nopeuksilla. Musiikin siirrossa kuitenkin mm. 128 kbps siirtonopeus on paremman äänenlaatunsa vuoksi lisääntymässä.



Kuva 6.4. Multimediyhteyksien pituuksien jakautuminen Washingtonin Yliopiston LAN-verkossa.

Kuvassa 6.4 on esitetty mitattujen multimediyhteyksien yhteyksien pituudet. Yleisimmät yhteysajat olivat 2,5 ja 4,5 minuutin välillä, mutta myös alle minuutin mittaisia erittäin lyhyitä yhteyksiä oli huomattavan paljon. Tämän perusteella käyttäjät suosivat lyhyitä

multimediaesityksiä. Toisaalta tuloksiin vaikuttaa tarjolla oleva multimediamateriaali. Esimerkiksi yksittäiset musiikkikappaleet tai elokuvien trailerit sopivat näihin aikarajoihin. Erittäin lyhyet, alle minuutin mittaiset pätkät puolestaan sopivat esikatseluun.

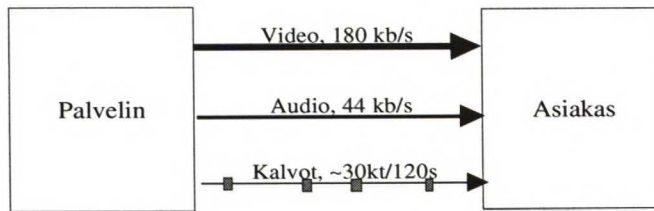
Edellä käsitellyt mittaustulokset kuvaavat multimedialiikennettä HTML-pohjaisessa WWW:ssä. Tämä selittää osaltaan matalanopeuksisten ja lyhyiden yhteyksien suosion. Siirtyminen paremmin multimediaa tukevaan multimedian kuvauskieleen tulee varmasti kasvattamaan erilaisten multimediasovellusten ja niiden aiheuttaman liikenteen määrää. Erityisesti pitkäkestoisten, laajakaistaisten multimediasovellusten osuus kasvaa, kun laadukasta videomateriaalia tulee verkkoon saataville.

6.3.2 Mallien kehittäminen

Simuloinneissa haluttiin keskittyä nimenomaan laadukkaan multimedian siirtoon. Mittaustuloksissa varsinaisten multimediasovellusten osuus oli erittäin pieni, joten mallia täydennettiin erilaisilla multimediasovelluksilla. Erilaisilla sovelluksilla pyrittiin kattamaan keskeisimmät multimedian käyttökohteet.

Reaaliaikaista multimediaa simuloitiin Outi Marttilan kehittämän etäopetussovelluksen avulla. Sovellus sisälsi videokuvan luennosta, luentokalvot ja keskustelupalstan. Luentokalvot vaihtuivat luennoijan kommentojen mukaan, mutta käyttäjät pystyivät myös itse selailemaan luentokalvoja. Luennon seuraamisen kannalta puheen ymmärtäminen oli ensiarvoisen tärkeää. Siksi sovelluksessa käytettiin erillistä kanavaa videolle ja audiolle. Samalla sovellus saatiin skaalautumaan paremmin erilaisiin verkkoihin. [53]

Etäopetussovellusta mallinnettiin simulaattorissa paketoimalla kaksi tasaista bittivirtaa tuottavaa generaattoria ja satunnaisesti kalvoja tuottava sovellus yhdeksi kokonaisuudeksi. Langattomassa ympäristössä on luonnollista keskittyä hyvälaatuisen audion siirtoon videokuvan kustannuksella, joten simuloinneissa käytettiin 44 kbps:n audiovirtaa ja 180 kbps:n videovirtaa. Luentokalvojen koko oletettiin tasaisesti jakautuneeksi 15 ja 45 kilotavun välille. Kalvojen vaihtoväli oletettiin Poisson-jakautuneeksi keskimääräisen välin ollessa 2 minuuttia. Sovelluksen toimintaa mitattiin audio- ja videoliikenteen saaman kaistanleveyden ja pakettikadon avulla.



Kuva 6.5. Etäopetuspalvelu.

Toiseksi sovellukseksi valittiin tilausvideotyyppisiä palveluja kuvaava videosovellus. Tilausvideossa valtaosa tiedonsiirrosta koostuu laajakaistaisen tilatun videon siirtämisestä, joten simulaatiomallissa keskityttiin nimenomaan videovirran mallintamiseen. MPEG-4 -koodatun videon keskimääräinen kaistaleveys on 500 kbps, huipukuormituksen ollessa 3 Mbps [44]. Puskuroitu tilausvideo voidaan tällöin tasoittaa 500 kbps:n levyiseksi CBR-liikenteeksi. Video siirrettiin UDP:n päällä, jolloin pakettikato on keskeisin tekijä palvelunlaatua mitattaessa.

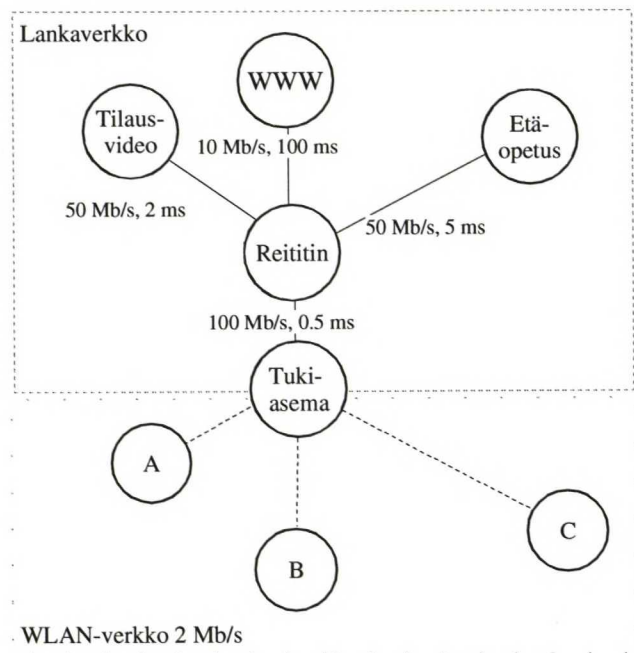
Verkon taustaliikenteen mallintamiseen käytettiin satunnaista WWW-liikennettä kuvaavaa mallia. Simulaattorin sisältämä pelkistetty HTTP-protokolla pohjautuu HTTP 1.0 -protokollaan ja siirtää WWW-sivut sarjallisesti yhden TCP-yhteyden kautta. Mallissa käytetyt WWW-sivut koostuvat yhdestä pääsivusta ja neljästä alikomponentista, jotka voivat olla esimerkiksi kuvia. Alun perin tarkoituksena oli käyttää satunnaismuuttujia kuvaamaan kaikkia liikenneparametreja, mutta tästä jouduttiin luopumaan, koska täysin satunnaisella WWW-liikenteellä suoritettujen testien tuloksia on erittäin vaikea analysoida. Sivujen ja komponenttien koko asetettiin kiinteästi keskiarvoon, jolloin WWW-liikenteen toimivuutta voidaan mitata yksinkertaisesti sivujen vasteaikojen avulla. Sivujen välisen mietintäajan oletettiin noudattavan Pareto-jakaumaa. Käytettyjen parametriarvojen ja jakaumien valintojen pohjalla on käytetty mm. SURGE:n, joka on realistista WWW-liikennettä mallintava liikennegeneraattori, piirissä tehtyä tutkimusta [54]. Simulaatioissa käytetyt parametrien arvot on esitetty taulukossa 6.4.

Suure	Satunnaisjakauma	Parametrit
Mietintäaika	Pareto	Keskiarvo 20, Huipukkuus 1.5
Pääsivun koko	Vakio	9357 tavua
Osasivujen koko	Vakio	9357 tavua
Osasivujen määrä	Vakio	4

Taulukko 6.4. Simuloidussa WWW-liikenteessä käytetyt tilastolliset mallit.

6.3.3 Simuloitava verkkoympäristö

Verkkoympäristöllä pyrittiin simuloimaan WLAN-verkon yhden solun toimintaa. Kuvassa 6.6 on esitetty simulaatioverkon rakenne. Verkko jakautuu lankaverkkoon, jossa multimediapalveluita tarjoavat palvelimet sijaitsevat, sekä langattomaan verkkoon. Langattoman verkon nopeudeksi valittiin 2 Mbps, joka on yleinen WLAN-verkoissa käytetty nopeus ja radiotien mallintamiseen käytettiin katvemallia, jonka parametrit asetettiin vastaamaan toimistorakennusta.



Kuva 6.6. Simuloitava verkko.

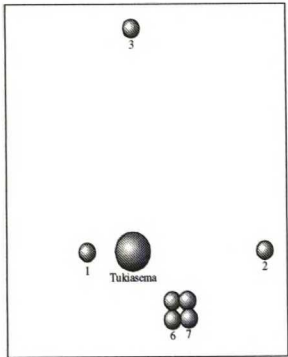
Tilausvideo sijoitettiin suhteellisen lähelle nopean yhteyden päähän, jolloin lankaverkon aiheuttamat kapasiteettiongelmat eivät pääse vaikuttamaan liikenteeseen. Viive asetettiin vastaamaan kotimaisen liikenteen tasoa ja kapasiteetti riittävän suureksi.

Etäopetuspalvelulla simuloitiin kotimaista, toisesta kaupungista tulevaa etäluentoa, joten palvelu sijoitettiin myös hyvätasoisien verkkoyhteyden päähän.

WWW-liikenteellä pyrittiin kuvaamaan lähinnä ulkomaille suuntautuvaa perinteisempää liikennettä, jonka pääasiallinen tarkoitus simulointien kannalta on ns. taustaliikenteen luominen. Kaistanleveys asetettiin suhteellisen pieneksi (10 Mb/s), koska ulkomaan liikenteessä verkon kapasiteetti ongelmat tulevat herkästi esiin. Suoritetuissa kokeissa ulkomaille suuntautuvien yhteyksien kokonaisviive vaihteli 50 ms ja 200 ms välillä, joten simuloidun WWW-liikenteen viiveeksi asetettiin 100 ms.

	X	Y
Tukiasema	0	0
Päätelaite 1	-10	0
Päätelaite 2	40	0
Päätelaite 3	0	100
Päätelaite 4..20	~30	~30

Taulukko 6.5. Tukiaseman ja päätelaitteen sijainti.



Kuva 6.7. Verkon kaavio.

Langattoman päätelaitteen etäisyys tukiasemasta vaikuttaa oleellisesti päätelaitteen saaman yhteyden laatuun. Erilaisilla etäisyyksillä toimivien päätelaitteiden keskinäistä vertailua varten päätelaitteet sijoitettiin verkossa erilaisille etäisyyksille. Vertailua varten kolme ensimmäistä päätelaitetta hajautettiin eri etäisyyksille tukiasemasta. Etäisyyksiksi valittiin 20, 40 ja 100 metriä. Loput päätelaitteet sijoitettiin noin 30 metrin päähän tukiasemasta. Kuva 6.7 havainnollistaa päätelaitteiden sijoittelua. Tukiaseman ja päätelaitteiden sijainnit on koottu taulukkoon 6.5.

6.4 Simuloinnit

Simulointeja suoritettiin kahdessa erässä. Ensiksi tutkittiin käyttäjien määrän lisääntymisen vaikutusta homogeeniseen liikenteeseen eri jonotusalgoritmeilla. Toiseksi tutkittiin yhdistelmäliikenteen käyttäytymistä eri käyttäjämäärillä ja CBQ-pohjaisen jonotuksen joustavuutta erilaisten luokkajakojen avulla. Yhteenveto suoritetuista simuloinneista on taulukossa 6.6.

Tapaus	Liikenne	Jonotusalgoritmit	Käyttäjien määrä
A1	WWW: 1 - 20	FIFO	1 - 20
		CBQ	1 - 20
A2	Video: 1 - 10	FIFO	1 - 10
		CBQ	1 - 10
A3	Etäopetus: 1 - 10	FIFO	1 - 10
		CBQ	1 - 10
B1	WWW: 8 Video: 2 x 500kbps	FIFO, CBQ	10
C1	WWW: 6 Etäopetus: 2 Video: 2	FIFO, CBQ	10
C2	WWW: 6 Etäopetus: 2 Video: 2	CBQ	10

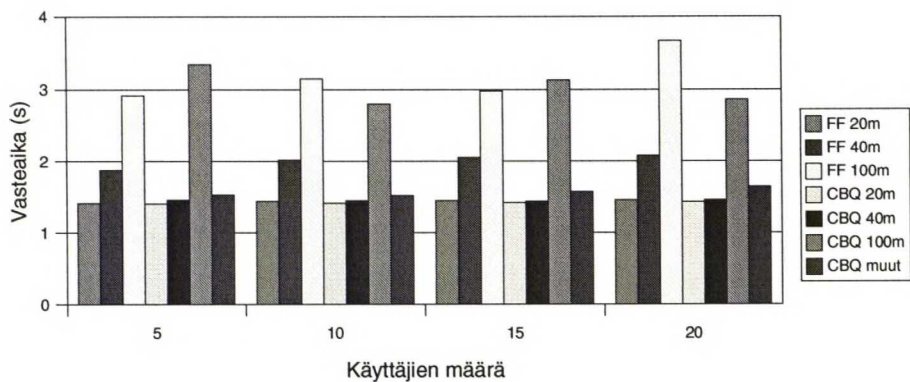
Taulukko 6.6. Suoritetut simuloinnit.

6.4.1 Liikennemallien perustestaus

Ensimmäisessä simulointisarjassa tutkittiin yksittäisten liikennemallien avulla generoidun liikenteen käyttäytymistä testiympäristössä. Tavoitteena oli luoda verrokkitulokset, joihin yhdistelmäliikennettä sisältävien simulointien tuloksia voidaan verrata. Tulosten avulla on myös mahdollista arvioida simulointimallien vastaavuutta todellisiin multimediasovelluksiin.

A1. WWW-liikenne

Ensimmäisessä simulaatiossa tutkittiin käyttäjien määrän lisääntymisen vaikutusta WWW-liikenteeseen. Käyttäjien kokeman palvelun laatua arvioitiin keskimääräisen sivujen latausajan avulla. Simulaatiossa verkkoon liitettiin viiden käyttäjän ryhmissä yhteensä 20 käyttäjää, jotka siirsivät edellä määritellyn liikennemallin mukaista WWW-liikennettä. Simulaatio suoritettiin sekä FIFO-jonotuksella että CBQ-jonotuksella. CBQ-jonotuksessa kolmelle ensimmäiselle käyttäjälle asetettiin muita korkeampi prioriteetti. Kuvassa 6.8 on esitetty kolmen ensimmäisen käyttäjän kokemat keskimääräiset sivunlatausajat eri käyttäjämäärillä. Kuvan neljässä palkkiryhmässä ensimmäiset kolme palkkia kuvaavat latausajoja FIFO-jonotuksella, kolme seuraavaa palkkia kuvaavat vastaavia aikoja CBQ-jonotuksella ja viimeinen palkki kuvaa alemman prioriteetin käyttäjien keskimääräistä vasteaikaa CBQ-jonotuksella.



Kuva 6.8. WWW-liikenteen vasteajat.

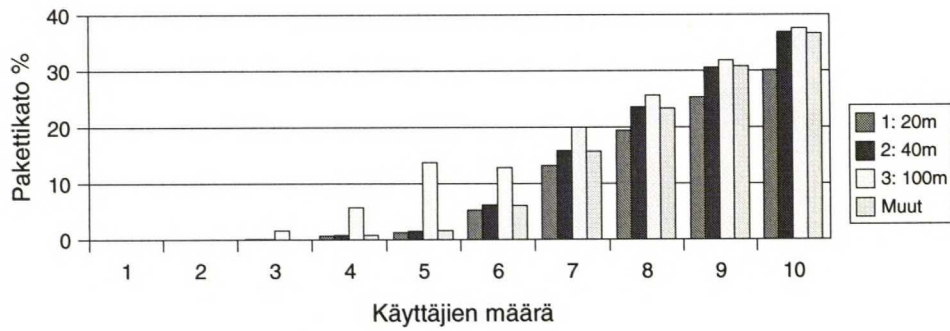
Tulosten perusteella voidaan todeta pelkän perinteisen WWW-liikenteen mahtuvan hyvin kulkemaan langattomassa verkossa, vaikka aktiivisia käyttäjiä olisi suhteellisen paljon. Käyttäjämäärän kasvu viidestä kahteenkymmeneen aiheuttaa ainoastaan hyvin marginaalista kasvua keskimääräisiin vasteaikoihin. Jonotusalgoritmilla ei tässä tapauksessa ollut merkittävää vaikutusta tuloksiin. Käyttäjän etäisyys tukiasemasta sen sijaan vaikutti käyttäjän kokemaan palvelunlaatuun. Absoluuttisesti mitattuna nämä muutamien sekuntien erot jäivät kuitenkin niin pieniksi, että niillä ei ole käyttäjän kannalta merkittävää vaikutusta palvelun laatuun.

A2. Videoliikenne

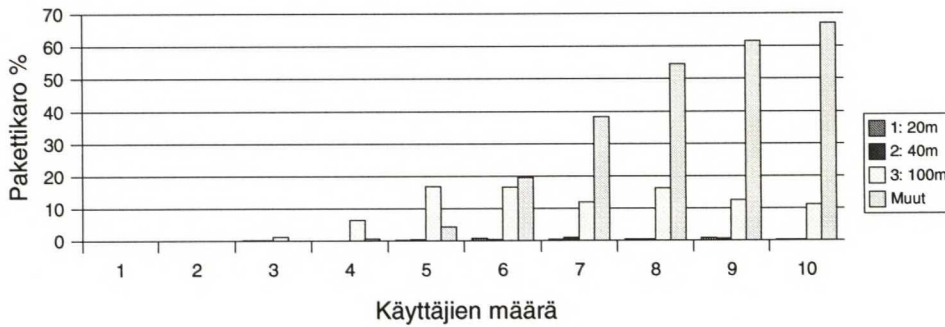
Seuraavaksi simuloitiin videoliikenteen siirtoa langattomassa verkossa. Simulaatiossa verkkoon liitettiin yksitellen uusia aktiivisia videopalvelun käyttäjiä ja havainnoitiin käyttäjien saaman palvelun tasoa pakettikadon perusteella. Käyttäjien tilaamien videoiden kaistanleveyksiksi valittiin ensimmäiselle käyttäjälle 500 kb/s ja muille 125 kb/s. Valitut kaistanleveydet vastaavat normaalikokoista ja neljännekstarckuuksista MPEG-4 koodattua videota. Käyttäjien ominaisuudet on koottu taulukkoon 6.7.

Käyttäjä	Liikenne	Etäisyys tukiasemasta	Muuta
1	Video 500 kb/s	20 m	Etuoikeutettu
2	Video 125 kb/s	40 m	Etuoikeutettu
3	Video 125 kb/s	100 m	Etuoikeutettu
4..10	Video 125 kb/s	~30 m	

Taulukko 6.7. Videoliikenteen käyttäjät.



Kuva 6.9. Videoliikenteen pakettikato FIFO-jonotuksella.



Kuva 6.10. Videoliikenteen pakettikato CBQ-jonotuksella.

Kuvassa 6.9 on esitetty kolmen ensimmäisen käyttäjän ja muiden käyttäjien keskimääräinen pakettikato FIFO-jonotuksella. Tuloksista voidaan havaita, että viidennen käyttäjän lisääminen nostaa kaukana tukiasemasta olevan käyttäjän pakettikadon yli kymmeneen prosenttiin, jolloin laatu ei ole enää hyväksyttävä. Samalla muiden käyttäjien pakettikato on ainoastaan muutaman prosentin luokkaa. Seitsemännen käyttäjän lisääminen nostaa myös muiden käyttäjien pakettikadon yli kymmeneen prosenttiin. Kymmenellä käyttäjällä, jolloin siirrettävää dataa on 1.6 Mb/s, kolmannes paketeista menetetään. Pienillä käyttäjämäärillä etäisyys vaikuttaa selvästi palvelunlaatuun, mutta ylikuormitustilanteessa sen osuus pienenee.

Vastaavan simulaation tulokset CBQ-jonotuksella on esitetty kuvassa 6.10. CBQ-jonotuksessa kolme ensimmäistä käyttäjää asetettiin muita korkeammalle prioriteetille. Korkeamman prioriteetin käyttöön annettiin 100% kaistanleveydestä. Tulokset osoittavat, että CBQ-jonotus pystyy pitämään lähellä tukiasemaa olevien käyttäjien palvelunlaadun erittäin hyvänä. Kaukana tukiasemasta olevan käyttäjän palvelunlaatu laskee kuitenkin alle hyväksymisrajan viidellä käyttäjällä, eli samalla hetkellä kuin FIFO-jonotuksellakin. Muiden käyttäjien palvelunlaatu alkaa laskea jyrkästi kuudennen käyttäjän kohdalla,

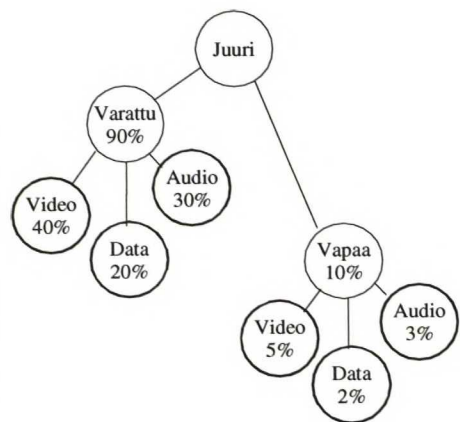
jolloin alemman prioriteetin luokka ylikuormittuu. Pakettikato nousee tällöin 20%:iin. Kymmenellä käyttäjällä kaksi kolmasosaa etuoikeudettomasta liikenteestä menetetään. Lisäkäyttäjät ainoastaan nostavat oman käyttäjäryhmänsä pakettikatoa, mutta eivät vaikuta etuoikeutettuun liikenteeseen.

A3. Etäopetuspalvelu

Etäopetuspalveluja simuloitiin edellistä testiä vastaavalla tavalla liittämällä yksitellen kymmenen palvelunkäyttäjää testiverkkoon. Palvelunlaatua mitattiin audio- ja videoliikenteen pakettikadon perusteella. Simulointi suoritettiin sekä FIFO- että CBQ-jonotuksella. CBQ-jonotuksessa käytettiin kuvan 6.11 mukaista luokkahierarkia. Vapaa kapasiteetti jaettiin Top-Level -algoritmin mukaan. Priorisointijärjestys on taulukon 6.8 mukainen. Kuvissa 6.12 ja 6.13 on esitetty audio- ja videoliikenteen pakettikato FIFO-jonotuksella ja kuvissa 6.14 ja 6.15 CBQ-jonotuksella. Kuviin on otettu mukaan vain kolmen ensimmäisen käyttäjän keskimääräinen pakettikato, sekä muiden käyttäjien keskimääräinen pakettikato.

Liikenne	Priorisointi	Maksimikaista
Varattu - Audio	1	30%
Varattu - Video	2	40%
Varattu - Data	3	20%
Vapaa - Audio	4	3%
Vapaa - Video	5	5%
Vapaa - Data	6	2,00%

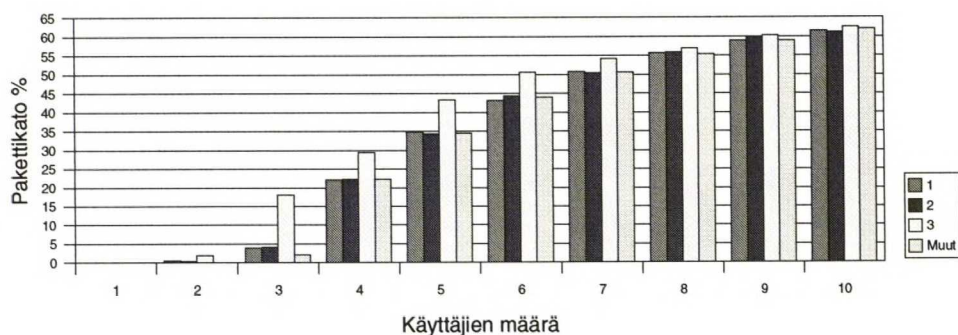
Taulukko 6.8. CBQ-liikenteen luokkajako.



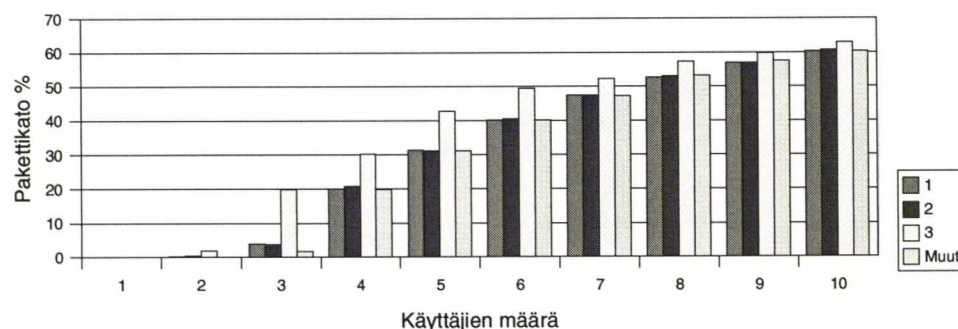
Kuva 6.11. CBQ-luokkahierarkia.

FIFO-jonotuksella keskimääräinen pakettikato ylittää 10% rajan neljällä käyttäjällä ja kasvaa kymmenellä käyttäjällä 60%:iin sekä audion että videon osalta. Selvästi etäopetusliikenne pystyy nopeasti tukkimaan langattoman verkon. Kuitenkin eri datavirroilla on eriarvoinen opetuksellinen arvo, joten datavirtojen priorisointi tarjoaa mahdollisuuden langattoman verkon käytön tehostamiseen.

CBQ-jonotuksessa audiovirroille annettiin ylin prioriteetti, koska audio on opetuksen seuraamisen kannalta kriittinen media. Video- ja datavirtoja voidaan pitää siihen nähden ylimääräisenä informaationa, joten niille annettiin alempi prioriteetti. Tulosten perusteella arvioitiin, että CBQ-jonotuksen ansiosta hyvälaatuinen audiovirta voitiin ylläpitää



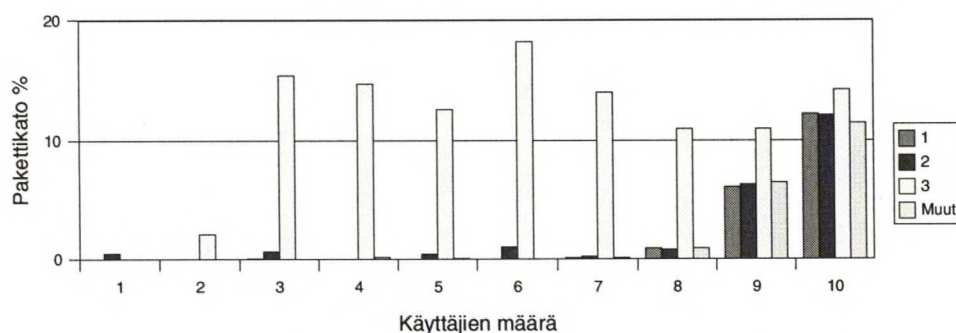
Kuva 6.12. Etäopetusliikenteen audion pakettikato FF-jonotuksella.



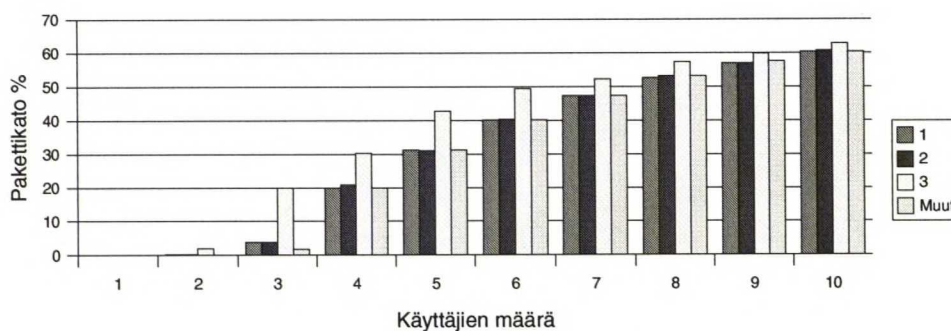
Kuva 6.13. Etäopetusliikenteen videon pakettikato FF-jonotuksella.

samanaikaisesti kahdeksalle käyttäjälle ja hyväksyttävän laatuinen yhdeksälle käyttäjälle. Käyttäjän etäisyydellä oli huomattava vaikutus audion laatuun. Kaukana tukiasemasta olevan kolmannen käyttäjän pakettikato audioliikenteessä nousee yli kymmenen prosentin jo kolmella käyttäjällä. Muiden käyttäjien pakettikadon perusteella voidaan olettaa, että myös kolmas käyttäjä olisi säilyttänyt hyvälaatuisen audioyhteyden, mikäli olisi ollut lähempänä tukiasemaa. Kymmenellä käyttäjällä kaikkien pakettikato nousee yli kymmenen prosentin, mikä ei enää ole audiolle hyväksyttävää. Pakettikadon nousu tässä vaiheessa johtuu siitä, että CBQ:n audiolle varaama kaista loppuu. Tällöin myös audioliikennettä joudutaan pudottamaan.

Videoliikenteen osalta CBQ-jonotuksella pystyttiin palvelemaan kahta käyttäjää kauempana tukiasemasta tai kolmea lähellä tukiasemaa. Videoliikenteen suuri kais-tanleveys syö sille varatun kaistan hyvin nopeasti ja neljällä käyttäjällä pakettikato nousee jo 20 prosenttiin. Koska audioliikennettä pystyttiin tarjoamaan 8-9 käyttäjälle, voitaisiin palvelua todennäköisesti tehostaa käyttämällä adaptiivista videota, jonka kaistaleveys laskisi verkon kuormituksen lisääntyessä. Tällöin voisi olla mahdollista yllä-pitää myös videoliikennettä samalle käyttäjämäärälle kuin audioliikennettä. Edelleen



Kuva 6.14. Etäopetusliikenteen audion pakettikato CBQ-jonotuksella.



Kuva 6.15. Etäopetusliikenteen videon pakettikato CBQ-jonotuksella.

aktiivisella CBQ:n luokkien muokkaamisella voitaisiin audion osuutta kasvattaa ja siten voisi olla mahdollista tarjota hyväksyttävän tasoista etäopetusta vielä useammalle käyttäjälle kuin kiinteillä jakosuhteilla.

6.4.2 Yhdistelmäliikenteen simulointi

Seuraavaksi pyrittiin tutkimaan eri liikennemallien vuorovaikutusta langattomassa verkossa. Yhdistelmäliikennettä testattiin kahdella testitapauksella. Ensiksi simuloitiin videoliikenteen ja WWW-liikenteen luomaa multimedialiikennettä ja toiseksi simuloitiin kaikkien kolmen liikennemallien yhteisliikennettä.

B1. Videon siirto WWW-liikenteen mukana

Yksi multimedialiikenteen perustapaus on videon siirto WWW-liikenteen joukossa. Tätä simuloitiin 10 käyttäjän verkossa siten, että kolme käyttäjää siirsi laajakaistaista videota (500 kb/s) ja loput käyttäjät siirsivät WWW-liikennettä. Testissä WWW-liikenteen vaikutusta korostettiin käyttämällä intensiivistä WWW-liikennettä, jossa sivujen lukeamiseen käytetty aika rajattiin 60 sekuntiin. Käyttäjien ominaisuudet on koottu taulukkoon 6.9.

Käyttäjä	Liikenne	Etäisyys	Aloitusaika
Video 1	Video 500 kb/s	20 m	0
Video 2	Video 500 kb/s	40 m	0
Video 3	Video 500 kb/s	100 m	2400
Http 1 - 6	WWW	~30 m	1200

Taulukko 6.9. B1 käyttäjien ominaisuudet.

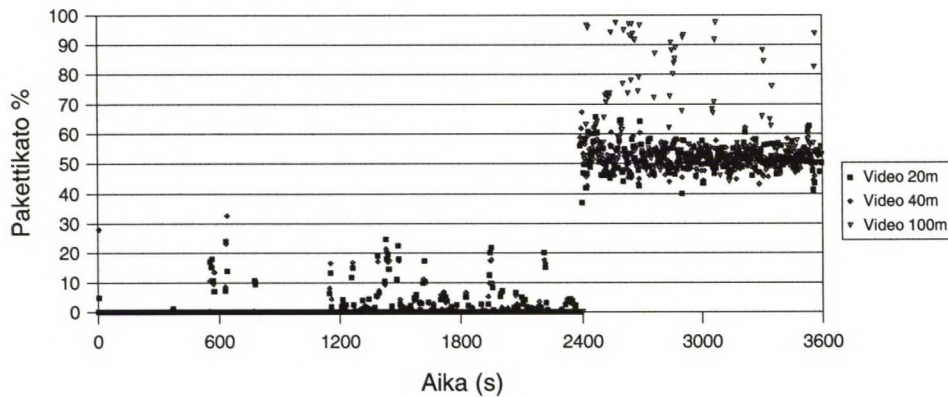
Simulaatio alkoi tilanteesta, jossa kaksi käyttäjää siirsi videota. WWW-liikenne tuli mukaan 20 minuutin kuluttua ja kolmas videoyhteys muodostettiin 40 minuutin kohdalla. Palvelunlaatua mitattiin pakettikadolla siten, että pakettikato mitattiin 5 sekunnin välein. Tämä loiventaa hieman lyhyitä purskeittaisia pakettikatoja, mutta antaa paremman kuvan keskimääräisestä pakettikadosta puskuroidulla liikenteellä. Tulokset on esitetty FIFO-jonotuksella kuvissa 6.16 ja 6.17 sekä CBQ-jonotuksella kuvissa 6.18 ja 6.19.

FIFO-jonotuksella havaittiin vain muutamia yksittäisiä pakettikatopiikkejä ensimmäisen 20 minuutin aikana. WWW-liikenteen alkaessa katopiikkejä alkoi tulla jatkuvasti ja erityisesti pienten, alle kymmeneen prosenttiin jäävien, piikkien määrä lisääntyi huomattavasti. Pakettikato jäi alle kymmenen prosentin 5 sekunnin tarkastelujaksoilla, joten videokuva säilyi pääosin hyväksyttävän tasoisena. Kolmannen videoyhteyden mukaantulo aiheutti ylikuormitustilanteen, joka nosti videoliikenteen pakettikadon noin 50 prosenttiin. Viiteen sekuntiin venyvät katkokset videoliikenteessä olivat yleisiä kaukana tukiasemasta. Samalla WWW-yhteyksien vasteajat nousivat muutamista sekunneista useisiin minuutteihin. Videoliikenteen aiheuttama raskas ylikuormitus tukki verkon kokonaan, jolloin kukaan käyttäjistä ei enää pystynyt hyödyntämään verkkoa.

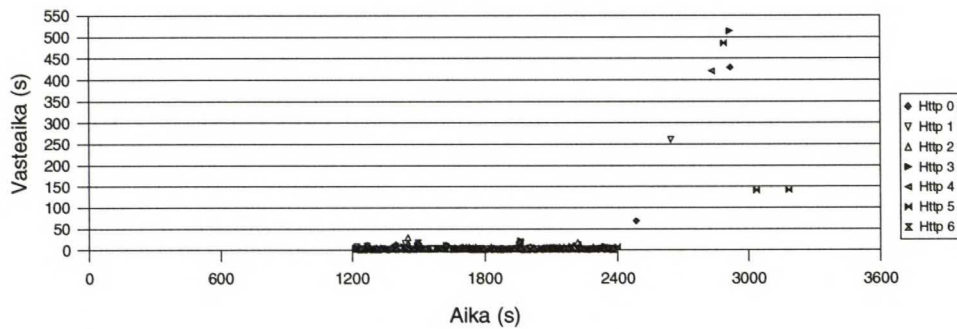
Videoliikenteen käyttämää kaistaa rajoitettiin CBQ-luokkien avulla. CBQ-luokkien parametrit asetettiin taulukon 6.8 mukaisesti. Videoliikenne jaettiin kahteen luokkaan siten, että yksi videoyhteys asetettiin etuoikeutetulle varatulle kaistalle ja kaksi muuta vapaaseen videoluokkaan. WWW-liikenne asetettiin alimpaan vapaa data -luokkaan. Käyttäjien liikenteen jako CBQ-luokkiin taulukossa 6.10.

Käyttäjä	Luokka	Kaistarajoitus
Video 1	Varattu - Video	90%
Video 2	Vapaa - Video	5%
Video 3	Vapaa - Video	5%
WWW 1 - 7	Vapaa - Data	2%

Taulukko 6.10. Video/WWW -liikenteen jako CBQ-luokkiin.



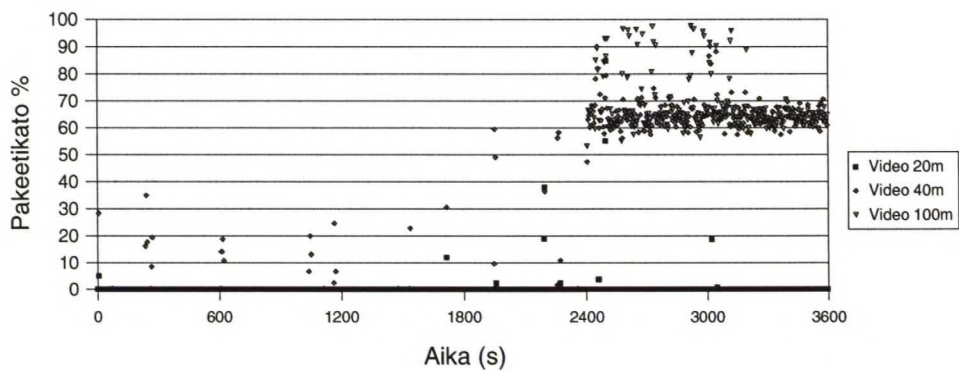
Kuva 6.16. B1-videoliikenteen pakettikato FIFO-jonotuksella.



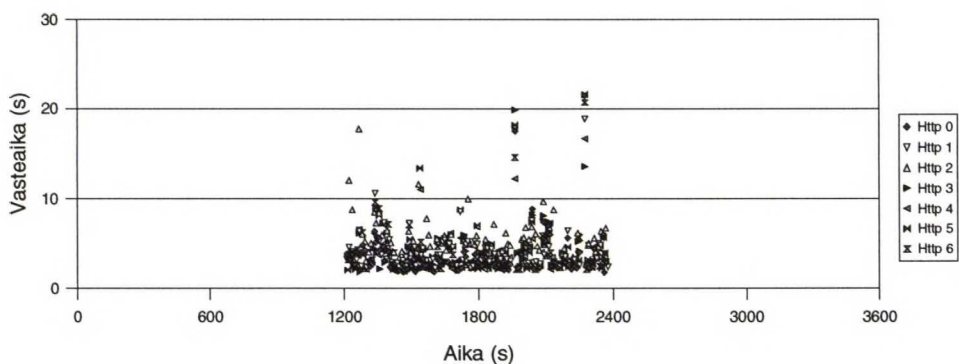
Kuva 6.17. B1 WWW-liikenteen vasteajat FIFO-jonotuksella.

CBQ-jonotuksen ansiosta videoliikenteen pakettikato pystyttiin pitämään yhden ja kahden videon tapauksessa pienenä. Ainoastaan yksittäisiä piikkejä pakettikadossa oli havaittavissa. Alhainen pakettikato saavutettiin WWW-liikenteen kustannuksella, jossa vasteajat kasvoivat 2 - 8 sekuntiin. WWW-liikenteen osalta tämä ei oleellisesti vaikuta palvelun laatuun. Kolmas videoyhteys kuitenkin tukki verkon vapaan kaistan ja esti sekä WWW-yhteyksien että vapaalle videokaistalle asetettujen videoyhteyksien toiminnan. Varatulla kaistalla siirtyvään videoon tällä ei kuitenkaan ollut vaikutusta.

Ylikuormitustilanteen aikana WWW-liikenteelle varattu kahden prosentin osuus kaistasta ei riittänyt ylläpitämään yhtään WWW-yhteyttä. Sen sijaan kaksi vapaalla kaistalla siirtyvää videoyhteyttä hukkasivat kaistaa erittäin suurella pakettikadolla. Selvästi varaa-mattoman kaistan käyttö videoliikenteelle aiheuttaa ongelmia verkon tehokkaan käytön kannalta.



Kuva 6.18. B1-Videoliikenteen pakettikato CBQ-jonotuksella.



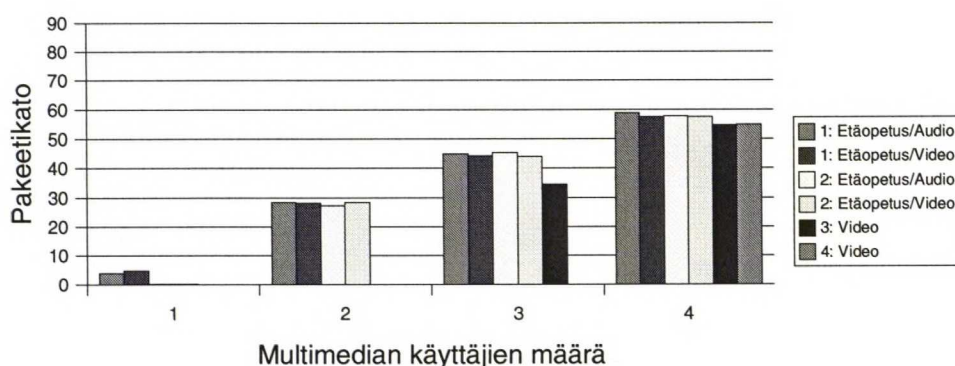
Kuva 6.19. B1 WWW-liikenteen vasteajat CBQ-jonotuksella.

C1. Yhdistelmäliikenne

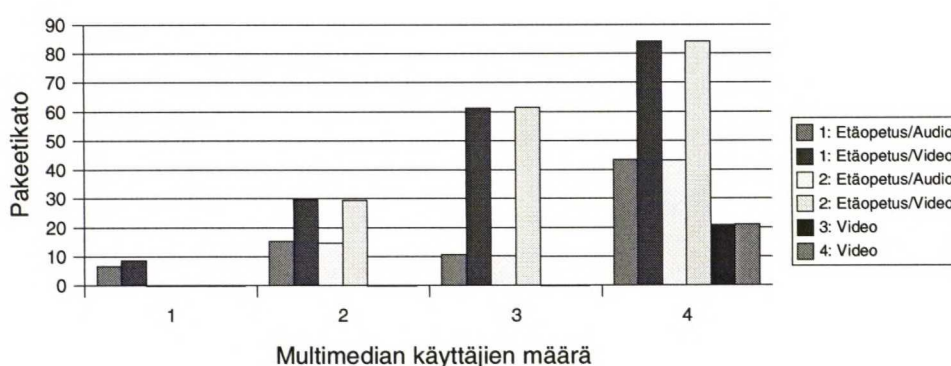
Yhdistelmäliikenteessä simuloitiin kaikkien kolmen liikenteen käyttäytymistä yhdessä. Simulaatio suoritettiin sekä FIFO- että CBQ-jonotuksella. Taulukossa 6.11 on esitetty käyttäjien käyttämät palvelut ja aloitusaajat ja luokat. CBQ-jonotuksessa käytettiin taulukon 6.8 mukaista luokkajakoa. Palvelujen laatua mitattiin pääasiallisesti pakettikadon perusteella.

Käyttäjä	Palvelu	Aloitusaika	CBQ-luokka
1: Video 1	Video	30 min	Varattu / Video
2: Video 2	Video	45 min	
3: Etäopetus 1	Etäopetus	0	Vapaa / Audio, Video, Data
4: Etäopetus 2	Etäopetus	15 min	
5-10: WWW 1 - 6	Web	0	Vapaa / Data

Taulukko 6.11. Yhdistelmäliikenteen parametrit.



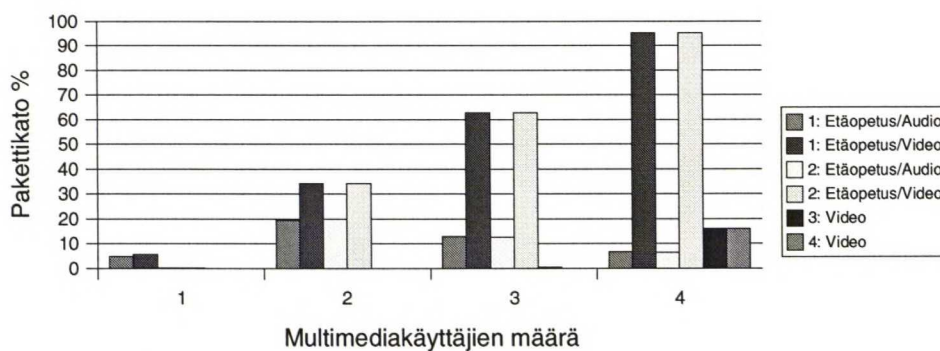
Kuva 6.20. Multimedian pakettikato FF-jonotuksella.



Kuva 6.21. Multimedian pakettikato CBQ-jonotuksella.

Simulaation tulokset FIFO-jonotuksella on esitetty kuvassa 6.20. Etäopetuksen siirto onnistui ainoastaan yhdellä käyttäjällä. Toisen etäopetuskäyttäjän mukaantulo nosti keskimääräisen pakettikadon audiolla ja videolla 30 prosenttiin, jolloin kaikki multimediaipalvelut tulivat käytännössä kelvottomiksi. Yksittäiset katopiikit nousivat usein 90 prosenttiin. Videoliikenteen mukaantulo simulaation puolivälissä nosti kaikkien käyttäjien katoprosentin 40 prosentin tuntumaan.

Luokkapohjaisella jonotuksella etäopetuksen audiokatoa pystyttiin kaventamaan kahden käyttäjän tapauksessa. Kuten kuvasta 6.21 ilmenee, nousi pakettikato audion osalta kuitenkin lähelle 15 prosenttia, jolloin audion käyttö ei enää onnistunut. Videokäyttäjän mukaantulo korkeammalla prioriteetilla vei kaistaa lähinnä etäopetuksen videolta, jonka pakettikato nousi 30 prosentista 60 prosenttiin. Samalla etäopetuksen audiokato laski odottamattomasti kymmeneen prosenttiin. Toisen videokäyttäjän mukaantulo nosti kaikkien käyttäjien pakettikadon yli hylkäämisrajan. Laajakaistainen korkean prioriteetin



Kuva 6.22. Yhdistelmäliikenteen pakettikato uusilla CBQ-parametreilla.

video kulutti nopeasti kaistan muulta multimedialiikenteeltä. Jäljellä olevan, tässä tapauksessa 10 prosentin, osuuden huolimaton jako eri palveluiden välillä teki kaikki multimediapalvelut käyttökelvottomiksi. Dynaamisella CBQ-luokkien kapasiteetin jaolla verkon yli siirtyvän liikenteen palvelunlaatua kokonaisuutena voisi olla mahdollista parantaa.

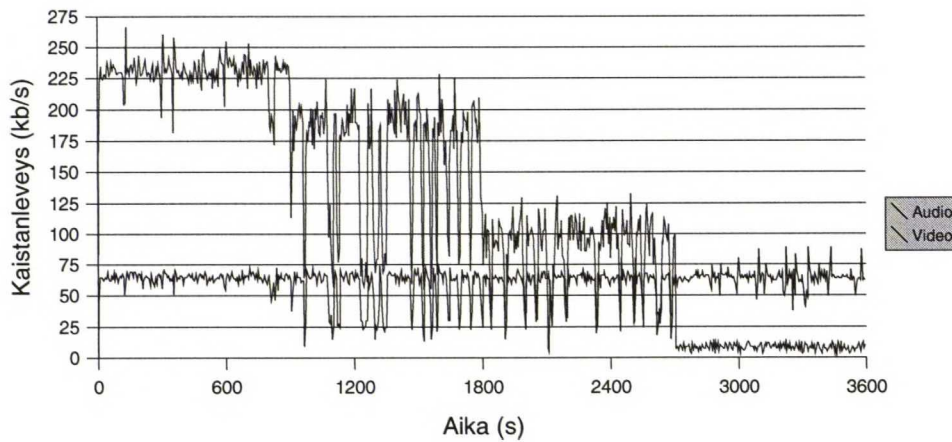
C2 Yhdistelmäliikenteen tehostaminen

Seuraavassa testissä vapaata kapasiteettia käyttävien palveluiden laatua pyrittiin parantamaan muuttamalla audion, videon ja datan keskinäistä suhdetta vapaan liikenteen luokissa. Erityisesti etäopetustyyppisiä palveluita ajatellen audion osuutta nostettiin videoliikenteen kustannuksella. Taulukossa 6.12 on esitetty uusi kaistajako CBQ-luokille. Muutoin testissä käytettiin edellisen C1-simulaation mukaisia asetuksia.

Liikenne	Priorisointi	Maksimi-kaista
Varattu - Audio	1	30%
Varattu - Video	2	40%
Varattu - Data	3	20%
Vapaa - Audio	4	7%
Vapaa - Video	5	1%
Vapaa - Data	6	2%

Taulukko 6.12. Uusi CBQ-liikenteen luokkajako.

Etäopetuksen audion siirron tavoitteisiin päästiin yllättäen vasta, kun kaikki käyttäjät olivat verkossa. Kuvassa 6.22 esitetyissä tuloksissa huomiota herättää se, että audiokadon määrä nousee kahdella käyttäjällä kahteenkymmeneen prosenttiin. Tästä kato laskee



Kuva 6.23. C2-simulaatio: Etäopetuksen kaistanleveys käyttäjälle 3.

tasaisesti, kun liikenteen määrää verkossa lisääntyy. Sama ilmiö oli havaittavissa jo edellisessä C1-simulaatiossa, tosin hieman pienempänä.

Ilmiön tutkimiseksi tarkasteltiin tarkemmin etäopetusliikenteen käyttäytymistä. Ensimmäisen etäopetusyhteyden audio- ja videoliikenteen kaistanleveydet on esitetty kuvassa 6.23. Kuvaajissa voidaan havaita videoliikenteen kaistanleveyden lasku ensin 900 sekunnin kohdalla, jolloin toinen etäopetusyhteys alkoi, ja seuraavan kerran 1800 sekunnin kohdalla, jolloin ensimmäinen videoyhteys käynnistyi. Lopuksi 2700 sekunnin kohdalla kaistanleveys laskee lähes nolleen, kun toinenkin videoyhteys tulee mukaan.

Välillä 900s - 2700s voidaan havaita useita voimakkaita kaistanleveyden lyhytkestoisia putoamisia murto-osaan alkuperäisestä. Vertaamalla tuloksia aiempaan A3-simulaatioon etäopetusliikenteellä, huomattiin käyttäytymisen vastaavan kyseisessä simulaatiossa kolmoskäyttäjän tuloksia. Kolmoskäyttäjän etäisyys tukiasemasta oli 100 metriä, eli selvästi pidempi kuin muilla. Tämän perusteella voidaan todeta, että katkokset liikenteessä johtuvat pääosin epävarmasta radiotiestä. Osasyllinen voi myös olla etäopetuksen sisältämä dataliikenne, joka aiheuttaa 30 kilotavun kuormituksen keskimäärin kahden minuutin välein.

Ylikuormitetussa verkossa fyysisen tason virheenkorjaus aiheuttaa yhden käyttäjän siirto-ongelmien näkymisen myös muulle liikenteelle. Mikäli radioverkkoon saapuvan liikenteen määrä on tarpeeksi pieni, niin tukiasema voi puskuroida paketteja uudelleenlähetyksen aikana ja uudelleenlähetyksiin käytetty aika ainoastaan viivästää muuta liikennettä. Ylikuormitustilanteessa tukiaseman vastaanottopuskurit ovat kuitenkin täynnä.

Koska tukiasema voi vastaanottaa paketteja ainoastaan samalla vauhdilla, kuin se pystyy niitä lähettämään, niin uudelleenlähetyksen aikana saapuneet paketit joudutaan pudottamaan. Luokkaperustaista jonotusta käytettäessä jokaisella luokalla on omat puskurit, joihin saapuneet paketit tallennetaan. Mikäli käytetään tavanomaista FIFO-puskuria, niin tällöin uudelleenlähetykset näkyvät pakettikatoina kaikissa niissä luokissa, jotka ovat täynnä. Muille luokille uudelleenlähetykset näkyvät ainoastaan viiveen kasvuna.

Simulaation tuloksissa tämä on selvästi nähtävissä. Alussa verkon kapasiteetti riittää piilottamaan yhden heikomman yhteyden päässä olevan etäopetuskäyttäjän tarvitsemat uudelleenlähetykset. Verkon ylikuormittuessa 900 sekunnin kohdalla videoliikenteelle varatut puskurit täyttyvät ja uudelleenlähetykset aiheuttavat videoliikenteeseen lyhyitä katkoksia.

Audio- ja videoliikenteellä näyttää kuvan perusteella olevan selkeä yhteys. Videoliikenteen ja audioliikenteen katkokset näyttävät osuvan samoihin kohtiin. Audioliikenteelle varattu kaistanleveys oli simulaatiossa 7 prosenttia 1.6 Mbps:sta eli noin 112 kbps. Kahden 64 kbps audiovirran lähettämiseen tämä on liian ahdas. Audioliikenne pystyy kuitenkin tarvittaessa lainaamaan kapasiteettia data-luokalta, joten suuren osan ajasta kapasiteetti riittää audioliikenteen siirtoon. Yksittäisiä pakettikadon arvoja tarkastelemalla voitiin vahvistaa, että simulaation loppuvaiheessa oli usein kymmenien sekuntien virheettömiä jaksoja. Suurin osa virheistä jäi muutamaaan prosenttiin, mutta keskiarvoa nostivat muutaman minuutin välein tapahtuvat voimakkaat katohiput. Katohipujen syy löytyy mitä ilmeisimmin radiotiestä, mikä jo aiemmin todettiin. Keskimäärin parin minuutin välein tulevat datapurskeet voivat myös olla osasyynä näihin katopiikkeihin. Datapurskeen aktivoituminen vaikuttaa kahdella tapaa. Ensinnäkin data vie kaistaa, joka muutoin olisi mahdollisesti voitu lainata multimedialiikenteelle. Toiseksi datapurske lisää liikenteen määrää etäällä sijaitsevaa käyttäjää kohti, mikä lisää radiotien aiheuttamien virheiden määrää ja uudelleenlähetyksien takia pudotettujen pakettien määrää.

Etuoikeutetun liikenteen määrän kasvaessa vapaalla kaistalla siirtyvä videoliikenne ei enää voi lainata varattua kaistaa, ja joutuu kaventamaan liikennettään. Tämä aiheuttaa heikon yhteyden päässä olevalle käyttäjälle suuntautuvan liikenteen vähenemistä ja samalla myös radiotien aiheuttamien virheiden määrän alenemista. Virheiden määrän lasku vähentää uudelleenlähetysistä aiheutuvia katopiikkejä ja parantaa audioliikenteen tasoa.

7 Johtopäätökset

Multimediasovellusten analysointi osoitti, että multimedialiikenteen siirrossa tärkeimmät laatuparametrit ovat kaistanleveys, pakettikato ja viive. Näiden vaatimusten täyttämiseksi tarkasteltiin erilaisia menetelmiä verkon palvelun parantamiseksi. CBQ-jonotus osoittautui lupaavimmaksi vaihtoehdoksi, jolla verkon palvelutasoa voitaisiin nostaa. CBQ-jonotuksen avulla langattoman verkon kapasiteetti voitiin jakaa osiin, joille tarjottiin erilaista palvelunlaatua. CBQ-luokkien avulla kaista jaettiin vapaaseen ja varattuun kaistaan. Nämä jaettiin edelleen videolle, äänelle ja datalle varattuihin aliluokkiin. Varatulle kaistalle annettiin 90 %:ia kaistanleveydestä ja vapaalle 10 %:ia. Luokille annettiin mahdollisuus lainata kaistaa toisiltaan. Tukiasema valvoi varatun kaistan käyttöä.

Luokkapohjaisen jonotuksen vaikutusta multimedialiikenteeseen tutkittiin simulaatiomallilla. Simulaatioissa mallinnettiin kahta multimediapalvelua ja WWW-liikennettä, joita siirrettiin yhden tukiaseman alueella. Simulaatioiden aikana käyttäjien määrä verkossa vaihteli yhdestä kahteenkymmeneen.

Simulaatioiden tulokset osoittivat, että CBQ-jonotuksen avulla on mahdollista tehostaa multimedian siirtoa langattomassa verkossa. Ylikuormitustilanteessa kaistan varanneet multimedialiikenneyhteydet toimivat normaalisti huolimatta muun liikenteen määrästä. Tämä kuitenkin edellytti, että kaistan varanneet käyttäjät pysyivät ilmoittamissaan kaistanleveyksissä, eikä varatun kaistan kapasiteettia ylitetty.

Käyttäjän etäisyydellä tukiasemaan oli selvä vaikutus. Kaukana (100 m) tukiasemasta sijaitsevalle käyttäjällä pystyttiin tarjoamaan hyvälaatuista palvelua vain silloin, kun verkon kuormitus oli alhaalla. Kuormitustilanteessa laatu heikkeni selvästi. Siirtovirheiden korjaamiseen käytetty aika kasvatti huonon yhteyden päässä olevan käyttäjän kuluttaman kaistan leveyttä. Korkeasti kuormitetussa palveluluokassa tämä aiheutti ylikuormitustilanteen, jolloin myös muiden käyttäjien liikenne häiriintyi.

Langattomissa verkoissa jaettu siirtotie yhdistää laatuparametrit toisiinsa. Langattoman siirtotien epävarmuus estää laatuparametrit ehdottomasti täyttävien palvelutakuiden antamisen. Heikon radioyhteyden päähän suuntautunut liikenne varatulla kaistalla voi vaarantaa palvelutason ylläpitämisen myös muilla varatun kaistan yhteyksillä. Vastaavasti heikon yhteyden liikennemäärän laskeminen voi nostaa palvelunlaatua kapeakaistaisilla yhteyksillä.

CBQ-luokkien parametreilla oli ratkaiseva vaikutus tuloksiin. Liian pienten osuuksien käyttö alemman prioriteetin WWW-liikenteelle ei estänyt sen nälkiintymistä. Vapaan kaistan käyttö multimediasyhteyksille osoittautui myös ongelmalliseksi. Ylikuormitustilanteessa laajakaistaiset palvelut tukkivat hyvin nopeasti vapaan kaistan verkosta. Esimerkiksi videoliikenteelle varattu viiden prosentin osuus kaistasta ei riittänyt kelvollisen videokuvan siirtoon, mutta jo siirron yritys tuhlassi kaistaa muulta liikenteeltä ja aiheutti kaikkien palveluiden katkeamisen vapaalla kaistalla. Yksi ratkaisu ongelmaan on vapaan kaistan varaaminen ensisijaisesti kapeakaistaisille yhteyksille.

7.1 Työn jatkaminen

Tässä työssä tutkittiin tilannetta ainoastaan yhden tukiaseman muodostamassa verkossa. Työn jatkamiseksi voitaisiin tutkia useamman tukiaseman muodostamaa verkkoa ja erityisesti liikkuvuuden aiheuttamia tilanteita.

Adaptiivisten CBQ-luokkien käyttö todettiin simulaatioiden yhteydessä yhdeksi mahdollisuudeksi parantaa tehoa langattomassa verkossa. Adaptiivisten luokkien tarjoama hyöty ja niiden toteuttaminen ovat kaksi mielenkiintoista jatkotutkimuksen kohdetta.

Simuloinneissa rajoituttiin vakiokaistaisten multimediasyhteyksien simulointiin. Adaptiivisten multimediasyhteyksien simuloinnilla voitaisiin tarkastella, kuinka paljon hyötyä ennalta varatulla kaistalla saadaan reaktiiviseen käyttöön verrattuna.

Työssä esiteltiin ratkaisu, jolla multimedialiikennettä voidaan tehostaa. Ratkaisu kuitenkin edellyttää, että on olemassa jonkinlainen protokolla, jolla tukiasema voi sopia varattujen luokkien käytöstä sovellusten kanssa. Tämän protokollan kehittäminen on välttämätön askel teorian siirtämisessä käytäntöön.

Lähteet

- [1] C.-H. Wu, D. Irwin. 1998. *Emerging Multimedia Computer Communication*. Upper Saddle River, NJ 07458. Prentice Hall. 443 s. ISBN 0-13-079967-X.
- [2] Information Sciences Institute, University of Southern California. 1981. Internet protocol. *IETF RFC 791*.
- [3] J. Postel. 1980. User Datagram Protocol. *IETF RFC 768*.
- [4] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. 1996. RTP: A Transport Protocol for Real-Time Applications. *IETF RFC 1889*.
- [5] IEEE. 1997. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. New York, NY. The Institute of Electrical and Electronics Engine. ISBN 1-55937-935-9.
- [6] Stardust Technologies. IP QoS FAQ [online]. 1999 [viitattu 24.1.2001]. Saatavissa: <http://www.qosforum.com/docs/faq>.
- [7] P. Ferguson, G. Huston. 1998. *Quality of Service, Delivering QoS on the Internet an in Corporate Networks*. New York. John Wiley. s. ISBN 0-471-24358-2.
- [8] R. Braden, D. Clark, S. Shenker. 1994. Integrated Services in the Internet Architecture: an Overview. *IETF RFC 1633*.
- [9] S. Floyd V. Jacobson. 1993. "Random Early Detection Gateways for Congestion Avoidance". *IEEE/ACM Transactions on Networking*. Vol. 1, nro 4, s. 397-413. ISSN 1063-6692.
- [10] T. Donald, M. May, J.-C. Bolot. 2000. Analytic Evaluation of RED Performance. *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. Tel Aviv, Israel, 26-30 March 2000. IEEE. s. 1415 - 1424. ISBN 0-7803-5880-5.
- [11] S. Floyd. 1994. "TCP and explicit congestion notification". *ACM Computer Communication Review*. Vol. 5, nro 24, s. 10-23. ISSN 0146-4833.
- [12] M. Shreedhar, G. Varghese. 1996. "Efficient fair queuing using deficit round-robin". *IEEE/ACM Transactions on Networking*. Vol. 4, nro 3, s. 375 - 385. ISSN 1063-6692.
- [13] J. Nagle. 1987. "On packet switches with infinite storage". *IEEE Transactions on Communications*. Vol. 35, nro 4, s. 435--438. ISSN 0090-6778.
- [14] A. Demers, S. Keshav, S. Shenker. 1989. Analysis and simulation of a fair queuing algorithm. *Symposium proceedings on Communications architectures & protocols*. Austin, Texas, USA, 1989. ACM Press. s. 1 - 12. ISBN 0-89791-332-9.
- [15] J. Bennett, H. Zhang. 1996. WF²Q: Worst-case fair weighted fair queuing. *Proceedings of Fifteenth Annual Joint Conference of the IEEE Computer Societies*. San Francisco, CA, USA, 24-28 March 1996. IEEE. s. 120 - 128. ISBN 0-8186-7293-5.
- [16] S. Golestani. 1994. A self-clocked fair queueing scheme for broadband applications. *Proceedings of INFOCOM '94*. Toronto, Ont., Canada, 12-16 June 1994. IEEE. s. 636 - 646. ISBN 0-8186-5570-4.
- [17] P. Goyal, H. M. Vim, C. Haichen. 1997. "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks". *IEEE/ACM Transactions on Networking*. Vol. 5, nro 5, s. 690 - 704. ISSN 1063-6692.

- [18] S. Floyd, V. Jacobson. 1995. "Link-sharing and Resource Management Models for Packet Networks". *IEEE/ACM Transactions on Networking*. Vol. 3, nro 4, s. 365-386. ISSN 1063-6692.
- [19] J. Wroclawski. 1997. The Use of RSVP with IETF Integrated Services. *IETF RFC 2210*.
- [20] B. Braden, et. al.. 1997. Resource Reservation Protocol (RSVP) - Version 1 Functional Specification. *IETF RFC 2205*.
- [21] S. Shenker, C. Partridge, R. Guerin,. 1997. Specification of Guaranteed Quality of Service. *IETF RFC 2212*.
- [22] J. Wroclawski. 1997. Specification of the Controlled Load Quality of Service. *IETF RFC 2211*.
- [23] P. White. 1997. "RSVP and Integrated Services in the Internet: A tutorial". *IEEE Communications Magazine*. Vol. 35, nro 5, s. 100-106. ISSN 0163-6804.
- [24] K. Nichols, S. Blake, F. Baker, D. Black. 1998. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. *IETF RFC 2474*.
- [25] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski. 1999. Assured Forwarding PHB Group. *IETF RFC 2597*.
- [26] G. Mamais, M. Markaki, G. Politis, I. S. Venieris. 1999. Efficient Buffer Management and Scheduling in a combined IntServ and DiffServ Architecture: A Performance Study. *1999 2nd International Conference on ATM*. Colmar, France, 21-23 June 1999. IEEE. s. 236 - 242. ISBN 0-7803-5428-1.
- [27] P. Bhagwat, P. Bhattacharya, A. Krishna, S. Tripathi. 1996. Enhancing throughput over wireless LANs using Channel State Dependent Packet Scheduling. *Proceedings of Fifteenth Annual Joint Conference of the IEEE Computer Societies*. San Francisco, CA, USA, 24-28 March 1996. IEEE. s. 1133-1140. ISBN 0-8186-7293-5.
- [28] C. Fragouli, V. Sivaraman, M. Srivastava. 1998. Controlled Multimedia Wireless Link Sharing via Enhanced Class Based Queueing with Channel-State-Dependent Packet Scheduling. *Proceedings of Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*. San Francisco, CA, USA, 29 March-2 April 1998. IEEE. s. 572-580. ISBN 0-7803-4383-2.
- [29] S. Lu, V. Bharghavan, R. Srikant. 1999. "Fari Scheduling in Wireless Packet Networks". *IEEE/ACM Transactions on Networking*. Vol. 7, nro 4, s. 473 - 489. ISSN 1063-6692.
- [30] World Wide Web Consortium. HTML 4.01 Specification [online]. 1999 [viitattu 27.4.2001]. Saatavissa: <http://www.w3.org/TR/html4/>.
- [31] Refsnes Data. Introduction to DHTML [online]. 2001 [Viitattu 25.4.2001]. Saatavissa: http://www.w3schools.com/dhtml/dhtml_intro.asp.
- [32] World Wide Web Consortium. Synchronized Multimedia Integration Language (SMIL 2.0) Specification [online]. 2001 [Viitattu 25.4.2001]. Saatavissa: <http://www.w3.org/TR/smil20/>.
- [33] Tom Lane. JPEG image compression FAQ, part 1/2 [online]. 1999 [viitattu 13.3.2001]. Saatavissa: <http://www.faqs.org/faqs/jpeg-faq/part1/>.

- [34] Motion Pictures Experts Group. 1993. Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s. .
- [35] J. Boyce, R. Gaglianella. 1998. Packet Loss Effects on MPEG Video Sent Over the Public Internet. *Proceedings of the sixth ACM international conference on Multimedia*. Bristol, UK, 1998. ACM Press. s. 181-190. ISBN 0-201-30990-4.
- [36] Motion Picture Experts Group. 2000. Information technology -- Generic coding of moving pictures and associated audio information: Systems. .
- [37] S. Gringeri, K. Shuaib, A. Lewis, B. Basch. 1999. Robust compression and transmission of MPEG-4 video. *Proceedings of the seventh ACM international conference on Multimedia*. Orlando, Florida, USA, 1999. ACM Press. s. 113-120. ISBN 1-58113-151-8.
- [38] R. Talluri. 1998. "Error-Resilient Video Coding in the IOS MPEG-4 Standard". *IEEE Communications Magazine*. Vol. 36, nro 6, s. 112-119. ISSN 0163-6804.
- [39] Apple Computer Inc.. Quicktime API Documentation: About Streaming [online]. 1998 [viitattu 4.4.2001]. Saatavissa: <http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/streamframe.htm>.
- [40] Ben Waggoner. Choosing a Streaming Video Technology [online]. 2000 [viitattu 4.4.2001]. Saatavissa: <http://www.dvwebvideo.com/2000/0500/waggoner0500.html>.
- [41] ITU-T. 1993. Recommendation H-261: Video Codec for Audiovisual Services at p x 64 Kbit/s. ITU Telecommunication Standardization Sector.
- [42] ITU-T. 1998. Recommendation H.263: Video codeing for Low Bit Rate Communication. ITU Telecommunication Standardization Sector.
- [43] T. Turletti, C. Huitema. 1996. RTP Payload Format for H.261 Video Streams. *IETF RFC 2032*.
- [44] F. Fitzek, M. Reisslein. MPEG-4 and H.263 Video Traces for Network Performane Evaluation [online]. 2000 [Viitattu 10.5.2001]. Saatavissa: <http://www-tnk.ee.tu-berlin.de/research/trace/trace.html>.
- [45] P. Alm, T. Nilsson, O. Hallnas, H. Kvålen. X Multimedia System [online]. 2001 [Viitattu 25.4.2001]. Saatavissa: <http://www.xmms.org>.
- [46] G. Karlsson. 1996. "Asynchronous transfer of video". *IEEE Communications Magazine*. Vol. 34, nro 8, s. 118-126. ISSN 0163-6804.
- [47] Y. Wang. Quality of Service in Mobile IP. 2000. Diplomityö. Teknillinen korkeakoulu, Tietotekniikan osasto. Espoo. 91 s.
- [48] L. Brelau, D. Estrin. 2000. "Advances in Network Simulation". *IEEE Computer*. Vol. 33, nro 5, s. 55-67. ISSN 0018-9162.
- [49] Monarch-projekti. CMU Monarch Project: Mobile and Wireless Networking [online]. 2000 [viitattu 31.11.2000]. Saatavissa: <http://www.monarch.cs.cmu.edu>.
- [50] T. Rappaport. 1996. *Wireless communications: Principles and Practice*. Upper Saddle River, NJ, USA. Prentice Hall. 641 s. ISBN 0-13-375536-3.
- [51] M. Gates, A. Warhavsky. Iperf version 1.1.1 [online]. 2000 [Viitattu 1.2.2001]. Saatavissa: <http://dast.nlanr.net>.

- [52] M. Chesire, A. Wolman, G. Voelker, H. Levy. Measurement and Analysis of Streaming-Media Workload [online]. 2001 [Viitattu 28.4.2001]. Saatavissa: <http://www.cs.washington.edu/homes/wolman/papers/usits-2001.pdf>.
- [53] Outi Marttila. XML Based Mobile Services. 1999. Diplomityö. Teknillinen korkeakoulu, Puunjalostustekn. osasto. Espoo. 75 s.
- [54] P. Barford, M. Crovella. 1998. Generating representative web workloads for network and server performance evaluation. *Proceedings of the joint international conference on Measurement and modeling of computer systems*. Madison, Wisconsin, United States, 1998. ACM Press. s. 151 - 160. ISBN 0163-5999.